DRAGON USER

The independent Dragon magazine

March 1988

Contents

Editorial

IF this *Dragon User* is late, then I'm on holiday; if it's on time, then I missed my 'plane, and if it's very, very late, then I'm in hospital somewhere in the Alps, hopefully suffering nothing worse than an overdose of spaghetti.

But seriously, there is a degree of controlled lateness here in an attempt to avoid the uncontrolled lateness which decended last time the staff took a week off. Don't blame the Postie till March.

The Postie may be to blame for the fact that we haven't got our *Miser's Dreams* yet. Roy is looking into it.

The Cardiff Airport Dragon show has been dogged by confusion, but if today is earlier than 27th Feb, there's still time to pack your bags and go. The enquiries number is on page 21.

This month we have a program to read PC (MS/DOS) discs onto Dragon discs, a report from the 6th 6809 Show, a hardware inverter, and all the regulars.

Even the Classified ads.

STOP PRESS+STOP PRESS+STOP PRESS
Just in case we're in time ... the Cardiff Airport Dragon Show really is on February 27th.

Telephone number
(All departments)
437-4343

Editor
HELEN ARMSTRONG

Production Editor
DRAGON EDITORIAL

Administration
CAROL FRITH

Advertisement Manager
DRAGON EDITORIAL

Marketing Manager
HELEN PERRY

Managing Editor
PETER KANE

Publishing Director
JENNY IRELAND

Subscriptions
UK £14 for 12 issues
Overseas (surface) £20 for 12 issues
ISSN 0265-177. Telex: 296275
Dragon User, 12/13 Little Newport Street, London WC2H 7PP
US address: c/o Business Press International, 205 East 42nd St, New York, NY 10017
Published by Sunshine Publications 1988

Typesetting and Production by Artset Limited, London NW1.
Printed by Headley Brothers Ltd. Ashford, Kent
Registered at the Post Office as a newspaper.
Dragon and its logo are trademarks of Eurohard Ltd.

How to submit articles

The quality of the material we can publish in *Dragon User* each month will, to a very great extent depend on the quality of the discoveries that you can make with your Dragon. The Dragon computer was launched on to the market with a powerful version of Basic, but with very poor documentation.

Articles which are submitted to *Dragon User* for publication should not be more than 3000 words long. All submissions should be typed. Please leave wide margins and a double space between each line. Programs should, whenever possible, be computer printed on plain white paper and be accompanied by a tape of the program.

We cannot guarantee to return every submitted article or program, so please keep a copy. If you want to have your program returned you must include a stamped addressed envelope.

Letters

This is your chance to air your views — send your tips, compliments and complaints to Letters Page, *Dragon User*, 12-13 Little Newport Street, London WC2H 7PP.

Space buggy

CONTRARY to popular belief, the 1.3 version of *Total Eclipse* is far from bug-free. Those who have got stuck with the game can contact me anytime on Great Missenden (02406) 2532. The first zero is not necessary when calling from overseas. There is an answering machine at my end so just hang on.

Julian Brown
Lavender Cottages
Salmons Lane
Prestwood
Bucks
HP16 0PY

Write to Pam

I AM still interested in producing software for the Dragon but am unsure as to what direction to take. Are there many more Dragon 32/cassette users than disc users? Are there many more multiple drive rather than single disc drive users? There are often requests for more use to be made of the 64's greater memory, but how many 64 vs. 32 users are there? *Dragon User* has never carried out the user surveys that other magazines feature from time to time. If readers would care to drop me a line detailing their current configurations and software usage/wants, not only will it help me enormously in my continuing support of the Dragon, but I will summarise the results (even if only one person replies!) for readers. It would also be of interest to know how many Dragon users who do not subscribe to *Dragon User* that readers know of.

Pam D'Arcy
21 Wycombe Lane
Wooburn Green
High Wycombe
Bucks
HP10 0HD

THERE are many reasons why *Dragon User* has not run a survey in recent years, most of them to do with the only source of editorial copywriting rushed off her feet, but some to do with the office administrator being rushed off her feet. Plus the fact that we get lots of opinionated letters (keep them coming). But since Pam has not only suggested one but offered to administrate it — we suggest you write to Pam and tell her your thoughts. She is a primary source of Dragon software.

Every month we will be shelling out a game or two, courtesy of our supplies, to the reader/s who send the most interesting or entertaining letters. So send us your hints and your opinions, send us your hi-scores and suggestions. Send us your best Dragon stories. What d'you think we are, mind readers?!

EXTRA PUFF

Professional advice

THROUGH the pages of DU, I know that quite a few of you are interested in the Dragon Professional computer. Well, quite recently, I wrote to GEC to find out more about the Professional, and here is a reproduction of the letter they wrote back:

Dear Mr. Linsley,

'Thank you for your letter regarding Dragon computers.'

'Unfortunately, we cannot help you with any information of the Dragon Professional, as we have only distributed the Dragon 32 and 64 models. We have not heard of the Professional version, and therefore we can only suggest that you contact *Dragon User* to investigate whether any of its readers have encountered this model. We can only assume that the model may have been made in small numbers as an unmarket prototype just before Dragon Data ceased trading.'

'I am sorry we cannot be of more assistance in this matter.'

'Yours sincerely, S. McManua, Service Technical Department.'

Well, GEC weren't very helpful, but how come GEC haven't heard of the Professional, when a few years back DU printed a picture of the Professional which had the GEC logo on the top of the machine?

David Linsley, 12 Glenn Barr, South Pelaw, Chester le Street, Co. Durham DH2 2JN

ALL true, all true. By the same token one might wonder: if *Dragon User* published that picture, how come they don't know about the Dragon Professional? But we don't. We have tried to get information before. I think that GEC are right, and that somebody worked on the Professional shortly before Dragon Data packed up. If someone had not told me, two years ago, that they had seen a Professional running, I would suspect the picture was of an advance prototype, ie a mock-up. It may have been so. It looks as though somebody at GEC did look into this, but once a project has been shelved and a few staff changes have taken place, it is almost impossible to find people who know what happened.

One thing is certain: there is no Dragon Professional on the market. But many thanks to Mr. Linsley for his effort.

Where's the chip?

IN the October 1987 edition of DU there was an article called *Better video*. This talked about the out of date MC6845, and the up to date NEC7220, which would give us better graphics and unlimited colours.

What it didn't say was where you can get the NEC7220 chip from, and when you have, how to instal it in the Dragon.

Duncan Chambers
30 Park Avenue
South Shields
Tyne and Wear
NE34 8QE

Routine righted

THANK you for printing my letter and program in the January 1988 issue (*Range right routine*). However, during the printing process some errors have been introduced. An apostrophe is missing before RIGHT in line 100. Line 160 should read IF P<=L THEN ... The variable is PASS in line 210. An open bracket is missing after MID$ in line 230. Line 260 should read IF PASS<=MS THEN B$=B$+ STRING$...In line 270 the word should of course be STRING$ and the GOTO 280 should in fact be a GOTO 220. Well, that's the lot. How is it that so many mistakes can appear in such a small program? Oh yes! One final thing ... perhaps you could have appended my name if not for self satisfaction then so that readers would know who to complain to.

Happy New Year.

Nigel Mason
31 Sidaway Close
Rowley Regis
Warley,
W. Midlands B65 9SJ

MISSING >s and <s usually mean somebody lost the original letter before proof-reading stage. Missing names tend to confirm this deduction. The fact that we can't find the letter just about sums it up. — Sherlock Sub.

Hi-score corner again . . .

I have seen the High Score corner in two issues of DU, so here are some of mine:
Shaolin Master — 100,200
Chuckie Egg — 193,040 — level 19
Speed Racer — 10 rules, 86,720
Shocktrooper — 52,250
I've been reading DU for two years, keep up the good work.

Nick Cutts
88 Hale Lane
Mill Hill, London NW7 3RT

UH oh. We just opened the can of worms again ...

I was reading September's high scores for *Chuckie Egg* and none of them are as high as the score my brother reached a couple of years ago. I don't remember the exact score, but it was just short of 750,000 on I think level 40.

Jeff Brook
46 Camroyd Street,
Dewsbury,
West Yorks WF13 1PE

I am fed up with looking at those pathetic high scores for *Chuckie Egg*. I am the champion, and that is final. I enclose a photograph of my second highest score of 919,470. I once topped my million, but had no film in my camera at the time.

I am a 39 year old mother of a Dragon user who has now confiscated *Chuckie Egg* from me, so that I have more time for cooking, cleaning etc.

Mrs. Sheila Stubbs (Basso)
Hill Cottage
30 Oakley Lane
Wimborne
Dorset
BH21 3AB

Don't let them turn Basso back into Brasso. Buy the kid a cookbook and yourself a copy of *Total Eclipse*. And now, for the entertainment of champions everywhere ...

We haven't printed the solutions to the Dragon crosswords yet, so here . . .

1	PITFIEND
2	OTHELLO
3	ELDIABLERO
4	SKRAMBLE
5	LASERZONE
6	MUTANTWAR
7	RAINBOWWRITER
8	DOWNLAND
9	COMPOSER
10	SHAOLINMASTER
11	DONKEYKING
12	JETSETWILLY
13	GHOSTATTACK

The December 1987 crossword solution.

1	ICECASTLES
2	ROBINHOOD
3	MUDPIES
4	SPACESHUTTLE
5	HUNGRYHORACE
6	TUBEWAYARMY
7	EIGHTBALL
8	CRAZYPAINTER
9	GRANDPRIX
10	DANGERRANGER
11	MOONCRESTA
12	DEVILASSAULT
13	SHOCKTROOPER

The January 1988 crossword solution.

I think I can claim to be *Chuckie Egg* master! I have had many times well over a million points — my record being just under 1,400,000 (level 79). As anyone else who may have got this far will know, this score is really irrelevant as I could easily beat it if I had enough time. After level 32, the game loops back to what was called level 25 (only it is now level 33). So it is possible to play for hours. My longest game lasted about 2) hours. But boredom takes over after the first couple of hours!

Andrew Lamb
2 Marsham Road
Hazel Grove
Stockport
Cheshire SK7 5JB

PS Are there any 6809 shows planned for around Manchester in the future?

THE last 6809 Show we heard of is being arranged in Cardiff for April 1988.

NEW HIGH-SCORE ON *CHUCKIE EGG*! 2,216,360 LEVEL 119 9TH AUGUST 1987 BY MARK AND JASON HENEGAN GAME LASTING ABOUT 2.45 HOURS..

THIS has been a public service announcement ...

FOR fear of strangulation from Ms. McCullough (September letters) I put forward my claim for the highest score ever on *Chuckie Egg*. I scored 4,081,790 on 24.10.84 and have rarely played it since (no wonder). After level 99, the numbers change to eggs and various other parts of the graphics of the game. I reckon I reached around level 232.

Even though folk may not believe me, this was definitely not poke aided and was on my trusty old Dragon 32. The game stops getting harder after about level 32 and stays at the same speed for infinity, so I could have scored an awful lot more, but after 8 hours almost continuous play I simply got fed up and went to bed! If anyone has beaten this score I'd very much like to see it.

N J MacArthur
Middlteon PO
Isle of Tiree
Argyll
Scotland
PA77 6XH

Noblesse oblige ...

I was very humoured at the high scores people send into high score corner. Paul's sister will have to strangle me because I got 6,750,000 and I had to kill all my lives because it was getting late, the level numbers changed into funny shapes and everything went really faster. The levels which got really hard were level 7 and level 8.

Lee Kalyan
19 Ampthill Road
Bedford
MK42 9JP

DOES this mean there's even less to do in Bedford of an evening than in the Isle of Tiree? Perhaps you need a foreign break. But ...

TO end all speculations about the highest score on *Chuckie Egg*: I do have it! It's 15,094,650!! This seems unbelievable, but is easy to explain: from about 450,000 the speed of the ducks stays constant, so you only have to train to get more bonus-men than you lose. I had enough after 32 hours, and interrupted the game in level 819.

With best wishes to all British Dragon users.

Andreas Verweyen
Rathenaustr. 9
D-8520 Erlangen
West Germany

WHAT we need is a totally new approach to this game...

REFERENCE hi score corner, you lot make me so envious with your multi-000s scores.

I have never got past the level which has three ladders at the top of the screen, each with an egg at the top.

How do you get the two eggs at the left hand ladders?

Anybody help me to get on and beat those massive scores!

H. Jones
54 Albion Street
Castleton
Rochdale
Lancs
OL11 2UL

No, No, H, I think you have the germ of a new concept here! All it needs now is somebody to work out a way of playing *Chuckie Egg* for 40 hours at a stretch without getting beyond level one.

If you have any new products for the Dragon — software or hardware — ring the News Desk on 01-437 4343

New adventure from Dragonfire

DRAGONFIRE Services have announced their first adventure for the Dragon, *Pyradventure*. Peradventure you were to enter the lost tomb of King Tutankhamun's father, you might find the ancient Amenophis III's gold death mask and his fabulous treasures. But will you come forth alive?

Pyradvenuture features real speech, freeze and save game facilities. "A real humdinger" say Dragonfire. The price is £3, plus 50p post and packing in the UK, £1.25 post and packing overseas. Overseas payment in pounds sterling by international money order, please.

Dragonfire have several other titles in the pipeline, including *Decathlon*, *Mazerace and Spellbox*, *Dragon Music* (series) and *Underlings of Croth*. Send an SAE for a full price list. Dragonfire are also interested in looking at software for prospective publication.

6809 show — Dates mixed up

IMPORTANT UPDATE: John and Helen Penn apologise for getting the dates muddled on their Show ad. in last month's issue, and plead a local 'flu epidemic and Benylin extract in mitigation.

The REAL dates of the upcoming shows are Saturday 30th April at Ossett Town Hall, one of the Dragon world's favourite venues, and Saturday 27th FEBRUARY at Cardiff Wales Airport. For further information, contact John Penn Software on (04203) 5970.

WP waits for words

R F Sibthorpe of Surrey has produced a new word processor to meet his requirements, and would like to know if other readers are interested in his program.

The vital criteria are a 64-character WYSIWINYNG (what you see is nearly what you get) hi res screen display, all necessary instructions for use displayed on the screen during run time, easy to use facilities for redefining printer codes for any connected printer and amend the dispslay of printer codes for easy comprehension, as well as user definable auto-repeat keys, user definable line end bell, auto word wrap, alterable line feed pitch, load/save to disc, on screen repositioning of blocks or windows, display with or without feint lines on screen, and, with the Epson FX100 printer, 32 font styles and sizes, fully mixable.

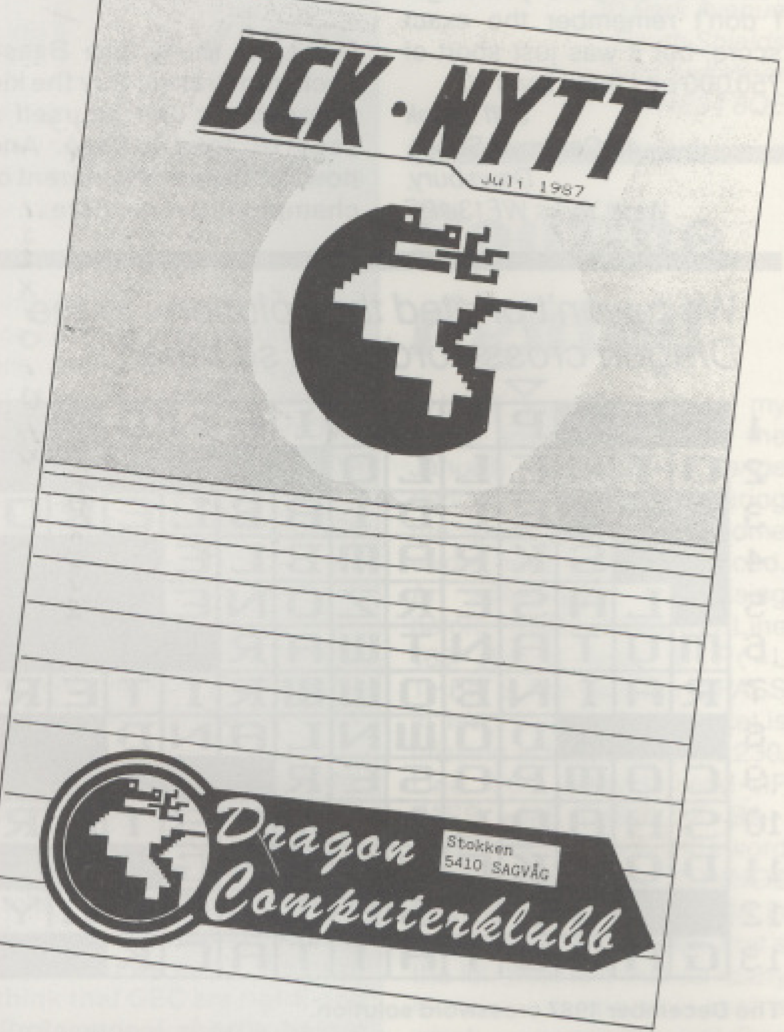
Anybody wanting further details can contact Mr. Sibthorpe at High Gren, The Drive, Belmont, Surrey SM2 7DH. It is not clear whether or not he is offering the package for sale — we get the impression that he is waiting to see if there is any interest.

North East User's club

THE North East Dragon Users Club has 14 members and meets every Wednesday at 8pm in the Etterick lounge of the Grindon Mill, Chester Road, Sunderland. Anyone who is interested is invited to go along, or contact the organiser, Chris Jobson, 23, Walsingham, Biddick Village, Washington, Tyne and Wear NE38 7HF.

68 Microcosm this month

THE December 1987 issue of *68 Microcosm*, the journal of the 68 micro group, includes OS968K on the Atari ST, the Transputer, a Flex disc formatter program, an introduction to the Data Protection Act, pick of topics from Tom Goodfellow's Mola bulletin board, small ads, lots of letters and the offer of a Cardfile program. Also a review of the 6th 6809 Colour Show. According to the membership sec., the last address DU printed was entirely wrong. What he means is that there was a typesetting mistake in the silly word in the middle. As this is no more ambiguous than the average person's handwriting, I imagine enquiries will still reach their destination. The entirely right address is Mr. K. Barnes, 174 Glen-Albyn Road, Wimbledon, London SW19 6HG.



Club in Norway

The Dragon Computerklubb of Norway has written to say that they will send a copy of their 24 page monthly magazine (A5, duplicated both sides, in Norweigan) *DKC-News* to anybody who enquires.

Dragon Computerklubb has been running for three years, and can organise orders of software from England for Scandinavian members. Enquiries to Ola Eldoy, Stokken, N-5410 Sagvag, Norway.

Computa — Text call

WOULD anyone who purchased a copy of *Computa — Text* from Dragonfire Services at the 6809 Colour Show in December please contact Dragonfire at the address below, sending their names and addresses, as the instructions are now ready.

Dragonfire Services, 13 Parry Jones Close, Blaina, Gwent NP3 3NH.

In the land of the crystal chandaliers

***Ken G. Smith** returns from the 6th 6809 Show with his impressions*

WHEN I saw there was going to be another 6809 Show in London, I didn't believe it. When I got to the Connaught Rooms, I believed it even less. Crystal chandeliers? CARPETS?? And casual seating — this was some venue for a Dragon Show. The Grand Hall seemed much larger than the Old Horticultural Hall. The exhibitors, who obviously felt safer with their backs to the wall, were spread around the perimeter. The new venue made this a very comfortable show to attend. The attendance figures seemed to me to be more in the hundreds than thousands and the size of the hall seemed to emphasize this.

The Dragon market is now so small that it is a pleasant surprise when new software is released. Well, just to prove that they can still surprise us, two suppliers chose to unveil new games at the show. The first of these, Quickbeam, put on their usual show with all their current range on sale plus the new one, *Indoor Football*. Truly a fascinating program with crowd noise and action replays, it is almost as much fun to watch as it is to play and it kept the younger show goers amused for hours. The second came from Pamcomms. Not previously noted for her games, Pam D'Arcy has come up with a real winner for her first try. Called *Grand Prix*, it is a split screen moter race game featuring several different tracks and the ability to race against the computer or another player. Available on disc or cassette, it provides real competition for *Speed Racer*. 'Well done, Pam' is all I can say.

Bus Extension

Harry Whitehouse, who you will remember used to trade as Peaksoft, was there with his usual supply of joysticks, those wonderful power supplies and some new keyboards. This time Harry, who jointly promoted the show with John Penn, was also offering his 'New Era' RS232 Interface which seems to be bringing a lot of Dragon 32 users on line. Offering a rom socket and a bus extention, is can be used in conjunction with a disc drive. However, it must be said, you will end up with a rather long cartridge. Although he is one of the few that will have to be told when to stop producing goodies for us, Harry is getting involved with other machines and was, in fact, knocking out Speckies at forty five pounds each.

A newcomer to our shows, Bill McGowan demonstrated how to use his *Printer Control* as a desk top publishing system. Also on show were the results of Bill's *Colourprint* program which converts the four colours of Pmode 3 into different densities of print. The end result is a black and white picture of amazing quality. McGowan's software has reached the stage where it outstrips anything of a similar nature and the best advice you can give to anyone who has just bought a printer is 'Chat to Bill McGowan'. Bob Harris, who also markets McGowan software, was demonstrating his own *KLIK* utility. This gives the humble Dragon 64 something of the environment of the Apple MacIntosh and runs under the *BASIC42* operating system. Bob's is one of the few software houses that is actually increasing their range of Dragon products. They even offer an upgrade service where for thirty five pounds they will boost your old 32 to a full 64K, making it capable of running BASIC42, FLEX and OS9.

> *It is well worth it to meet old friends, make new ones, and talk about the machine to which we are all committed.*

H C Andersen's arrival proved that stories of his fairy tale exsistence were nothing more than a Grimm rumour. The big blonde Dane was knocking out games at a pound each and OS-9 at under eighty pounds, complete with both manuals. He certainly seemed to be enjoying himself. I only hope he found it a rewarding experience.

With mountains of suplus software slowly shrinking away John Penn has shifted his emphasis away from selling off surplus stock at discount prices and is now distributing more current software. For those of you who like me were wondering what has happend to the final bug free version of *Total Eclipse* I can tell you that John has it. He is also distributing David Maken's excellent *Picture Maker*. David was at the show demonstrating his wares and I must say the results were remarkable.

All the usual user groups attended plus one new one dedicated to Delta Dos users. (DUDE). They had one of the new Tandy 128K CoCos with enhanced graphics on show. Rumours were rife about this machine, mainly as to whether production was on or off. The performance looked quite good but whether it could stand against competition from the likes of Atari is another matter. DUDE were up on the balcony along with the boys from 'Dragon's Roar' magazine. Their supply of free copies soon ran out and Simon Jones was taking orders. I have now received my copy and have to say that it is very professionally presented and caters for a wide variety of readers. Also up on the balcony, I had a very interesting chat to Ted Baccarelli of 68 Microcosom. If any of you wants to run hard discs on a Dragon then Ted is your man. He can probably even supply the parts.

Well, as usual, I enjoyed my time at this year's London show. However, the falling attendances make me wonder how much longer such an event can continue in its present form. As long as it does, I, for one, will gladly pay for my ticket. It is worth it to meet old friends, make new ones and talk about the machine to which we are all committed (or for which we should all be committed.)

"Stirring" Smith

Now, to stir things up bit, I have decided to make the following nominations for an imaginery show award called the Golden Digit. 1. Best new hardware. Harry Whitehouse for his 'New Era' RS232/Modem driver.

2. Best new software. I have to share this one between Pamcomms for *Grand Prix* and Quickbeam for *Indoor Football*.
3. Best show idea. *Dragon's Roar* for their live Micronet link up.
4. Best bargain. National Dragon User Group for their thirty five pound Modem package.
5. Man/Woman of the show. Hans Andersen for his enthusiasm and the effort he put in just to get there.
6. Wooden Raspberry. Compusense for being the only people offering less than ever. Will they even turn up when they have sold all their 'Plus' boards?

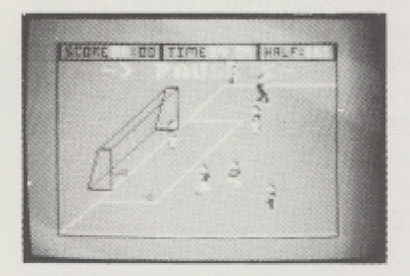
Dragonsoft

Getting into this game is just for kicks

Title: *Indoor Football*
Supplier: Quickbeam Software, 36 Salisbury Road, Hoddesdon, Herts EN11 0HX
Price: £8.95

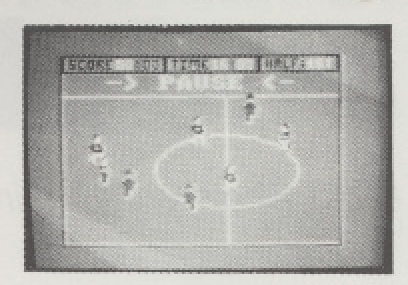
I seem to have become *Dragon User's* football correspondant as our dear Editor has given me the task of reviewing Quickbeam's latest offering...*Indoor Football*.

There seems to be no shortage of quality Dragon football games around, what with *Champions* (and the new *Champions*), *Football Manager* and *Crazy Foota*. But now Quickbeam have come up with what is quite simply the best 'full perspective arcade style football game on the Dragon'. These words come from Quickbeam's advert for the game, but echo my views entirely. Sure the rest are good, but *Indoor Football* is better.



Indoor Football follows the same method of play as *Crazy Foota* (where the player controls his/her team and battles it out with an opponent). I prefer this type of football game to the likes of *Football Manager* where there is absolutely no skill involved, and the result of each game depends really on luck. However, I don't think it's quite right to drop all the features of that type of game; one of the better features of *Football Manager* and *Champions* was that each team played in a league and throughout the season battled for a place in the Cup and Europe. In *Indoor Football*, one simply plays either the computer or an opponent for a certain time and once the time is up, that's it.

In the game you control the flashing character (you'll notice there are two. You control the one in your colours). You have direct control over this player, and with him it is possible to tackle another player, simply run up the field



with the ball, pass the ball to another player, or take a shot at goal.

There are other players on the field which move independently, and they are capable of running towards a loose ball. Once you've got the ball it's a hard slog to get to the other side because there is always an opponent hot on your heels and you've got the ball, you're slower than the other player. What it does mean is that you frequently have to run round in circles trying to dodge your opponents player. However if you manage to beat this player and the rest of his team you'll just have goalkeeper to contend with, and assuming you score you will be awarded with a loud cheer and a slow motion action replay.

The game is split into two halves, the ball being replaced on the centre spot at the beginning of each half.

Before the game begins you are given a number of choices. You can choose the number of players in each team, you can choose whether to play a friend or the computer and you can select the length of each game.

One thing which struck me after loading was the lack of title music. As most of you will be aware, Quickbeam have produced some great sounds on their games with their DAMS program (must get it some day). However there's no music on this game and indeed there is very little sound on the game, the noise of the ball bouncing and the cheers



from the invisible crowd being the only sound.

The graphics are in 3D and are good (if a little simple). The players seem very familiar to me — they're very much like the soldiers on *Fire Force* and they too are drawn well. As usual Quickbeam have used larger than normal sprites to draw the men.

There seems to me to be a slight fault in the detection, as all too often I've had goals scored against me that are quite obviously not valid. Even in slow motion it's obvious that a goal should not have given. The design of the goalposts don't help the matter very much either as they're in 3D, making it even more difficult to distinguish just what is a goal.

Another problem which happend quite often was that although I had picked 7 or 8 players, after a while most of them disappear — in fact I've seen many disappear before my very eyes. (*Transferred for a large fee, very likely — Ed.*) Quickbeam have come up with some great cassette inlays but this time their presentation is amateurish to say the least. Even *Crazy Foota* seems to be more professional.

Now I know many of you will say it's the game that counts, not the cassette inlays, but if computer enthusiasts who sell their games go to the bother of making their product look decent enough then I'm sure Quickbeam could do the same.

If these few commants lead you to the impression that I loathe this game, you've been fooled! I absolutely love this game, the good graphics coupled with the extreme realism of this game combine to give possibly one of the games of '88. OK, it may not be graphically stunning, it may not have hot sound, but after a playing *Indoor Football* you'll toss all that to one side, and get on with playing what is a great game.

For me it was a joy to play *Indoor Football*, it was certainly addictive and very competitive. I'll give it five Dragons, but if you don't agree you know where to find us.

Donald Morrison



Living on

Title: *Zotoka*
Price: £2.99
Supplier: Preston

CIVIL war, hurricane and invading armies, all part of daily life on the newly found island of Zotoka. Having accidentally discovered the isle you get the job of running it, and you might guess that it's not one of the most secure jobs in the world.

Zotoka is almost two programs; on side A of the cassette are instructions with what seems like several dozen pages of text. These instructions delve to great depths to help you understand just what is going to happen — but after so many games where instructions have consisted of something like 'Type Cloadm and the game will start automatically. Use right joystick', these attempts have to be welcomed.

The instructions even go to the extent of giving you a little test to see how much you've understood. If you answer correctly you move to further instructions. If you are nearly right you are told of the error of your ways and then move on. However, if you type 27 when the answer is 3 you're told not to be silly and answer again.

The length of time I've spent describing the instructions shows their size but the game is not as complicated as it may momentarily seem. You are given a term of office ranging from 5 to 20 years starting in 1987; then a skill level 1 is easy, 3 is hard and 2, well it hasn't a title, middling I suppose. You're also asked if your trusty Dragon will permit the 65495, 0 poke to help speed up things slightly.

The little Southsea paradise consists of 16 blocks of sea and 38 of land making up a 9 by 6 grid. Given a starting population of 500 and food for that number, the task with a limited supply of money is to meet the growing population.

New software for review should be sent to *Dragon User*, 12-13 Little Newport Street, London WC2H 7PP.

Heir raising tasks

Title: *The Heir of Tyos*
Supplier: Simon Hargrave
Price: £5.00

SIMON Hargrave's adventures, starting with *Starcrash*, a game with a good plot, which I rated at four Dragons in the Dec '87 issue, have by the fourth in the series been enhanced tremendously so that the end product is even more of a professional all-round package.

In *The Heir of Tyos* in the tradition of the hero having an odd name, you are cast as Krantz Menostu and live on planet Tyos in the Xenopho system. The task is to find the amulet of Tyos to stop a neighbouring King's siege.

You start, when you awake at the top of a pyramid shape base by a teleporter and the number of locations increase as you progress down the floor levels of the pyramid. In total there are over 200 locations, although despite my best and drastic attempts I have absolutely no idea how many.

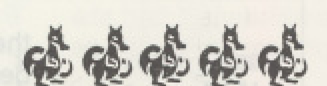
Other locations include medical laboratories, where a nurse ominously lurks with a syringe, an apartment, covered balconies and a cafeteria. Plus there's a shattered tube' where vacumn no longer exixts and you are promptly sucked to your death.

The reason for my scepticism over any task being simple in this adventure is that instead of merely finding an object and next game going back there you have to search other areas for the item. Not being able to boycott previously unfruitful locations is due to the items are randomly placed.

Complex could be used to describe the commands, 'Give the syringe to the nurse after removing the needle' being reminiscent of *Juxtapositon*.

This is easily the hardest of the series, so give it a try.

Philip Stott



an island

Each year you can buy items like crops or housing, one unit of each being sufficient for 500; don't buy enough and the natives will get restless. Also on the shopping list are various priced items like factories which provide work and earn extra money (watch out for Made in Zotoka labels), schools, hospitals and others also help keep the masses quiet for a while.

A small island is easily attacked though and so a defence is needed. For 75 units of money 50 soldiers are added to the starting number of 150. These men are certainly needed when foreign armies start attacking because if you allow 3 of their bases to be set up permanent redundancy looms and the guillotine will be wheeled out for you.

Civil wars are perhaps even worse because due to the unrest, the masses rampage through the streets and fields destroying most of the hard work and effort.

Hurricanes have a similar effect, there's one at least every seven years (sounds like the bus service) which starts and travels randomly destroying all in its path (perhaps it is the bus service!)

And that's about it. The games displayed on what has become almost the norm — that is the black/buff screen, the island displayed graphically with redefined text messeages underneath. If you survive your term you win if you don't you lose. There is a score feature but this is almost irrelevant in a live or die game.

Not a brilliant game, or for that matter original but it is excellent value and is well programmed and displayed. Definitely one for the trainee meglomaniacs.

Philip Stott



Put another disc on the juke box, baby

Title: various
Supplier: Preston
Price: £4.99 each Dragondos disc

A new range of budget priced disc games has been launched by R. & A.J. Preston. The games are not new ones but this is the first time they have been available on disc at the low price of £4.99. Two game discs are reviewed here; the first is a compilation of three games, *Ruby Robba*, *Perilous Pit* and *Desperado Dan*.

An initial menu is displayed after BOOTing the Dragondos disc. From this you can choose to play any one of three games. It's not possible to go back to this menu; to change games requires powering off.

'*Ruby Robba* is a curious mixture of the *Connect 5* and *Sliding Square* puzzles. Using either keyboard or joystick you control a pointer which moves around the edge of a 10 by 10 grid of blue blocks. These blocks are either blank, or contain snakes or guards or a mystery bonus of between 100 and 300 points. The ruby is shown as a red block. Within the blank squares are mines which can to kill the snakes and guards for extra points.

The blocks can be moved bi-directionally using the fire button or space bar. The object of the game is to push the square containing the ruby into a 'trap' at the top left of the screen. This is made more difficult than it sounds by the snakes and guards, which will return the ruby to the bottom right of the screen if they get near it.

Although this initially seems quite an original game, the challenge can fade — there isn't really enough variety in the total of 15 difficulty levels. The object of the game is simple to grasp, but to capture the ruby requires some logical thinking — hence this game is ideal for the younger player.

Desperado Dan is a familiar platforms and ladders type game — guide the little figure down the screen avoiding the flying bits, ducking the falling bits and jumping the stationary bits. *Desperado Dan* is not one of the classics of its genre.

Perilous Pit is a simplified version of *Donkey Kong*. The slight difference is that diamonds must be collected one at a time and taken to the truck at the top of the screen. Three diamonds are needed to move onto the next screen and there are four screens in all. An 'Orbis' appears on each screen (it looks like an old-style radio to me!) which slowly homes in on you once you start jumping. Moving platforms and other nasties must be negotiated.

The mode 3 graphics are fairly well defined, but do tend to get rather flickery at times (is this game machine code or just compiled Basic I wonder?). If you specifically want a version of *Donkey Kong* then stick to Microdeal's offering. As a third on this budget disc *Perilous Pit* is a nice clone.

None of the games on this disc are exactly going to set the world on fire. However, as a set of three, and at a price of £4.99 they can be recommended to the gamesters who can't wait for tapes to load.

The second disc I looked at contains just one game; *Kung Fu — The Master*. Again this is loaded by BOOTing the Dragondos disc. The object of the game is to rescue your girlfriend (isn't it always?) who is, for some unexplained reason, being held captive. To do this you must get past screen after screen of 'trainees' (who look just like you), knifemen, snakes, bombs and fire-breathing dragons. The joystick is used to do all the usual kung-fu stuff like run, jump, duck, foot sweep and various types of kick — all nice and violent! Points seem to be scored for hitting just about anything in sight.

The mode 3 graphics are quite well animated and flicker free with some nice horizontal scrolling included. The baddies are also nicely defined.

This is probably one of the better kung-fu games around for the Dragon at the moment.

Brian Cadge



Pamcodes

Pam D'Arcy continues her introduction to machine code

HOW far did you get with yellow blob? With my experience (grey hairs and all), I could see a number of problems from the outset. However, I shall try and put myself in your shoes — or at least, try and give you footsteps to follow in. My first move on a step by step basis, was to type in the source as it stood and attempt to assemble. This is now where half of this article may not be relevant to you as your assembler may have accepted and assembled it all quite correctly (and I apologise for a just spotted wrong value in the comment column of line $5023 — it should have read ;27 FB). For those with similar assemblers to mine, four lines of code were rejected — where there were values in the operand column that contained either of the hexadecimal letters A-F (representing decimal values 10-15).

```
5001                 * LISTING 7
5001                 *
5001                 * YBLOB7 (FILENAME)
5001                 *
5001                 * THE YELLOW BLOB - PAGE 56
5001                 * FROM "DRAGON MACHINE CODE"
5001                 * BY JONES & COWSILL (SHIVA)
5001                 *
5001                 * TYPED IN AS PER THEIR LISTING
5001                 * EXCEPT FOR ; PRECEDING COMMENT
5001                 * DATA AND $ INSERTS TO OBTAIN
5001                 * AN ERROR FREE ASSEMBLE
5001                 *
5001                 * USING DSKDREAM ASSEMBLER
5001                 * AFTER CLEAR200,&H5000
5001
5001 8E0190     6404    LDX   #0400  ;8E 04 00
5004 BF1900             STX   6400   ;BF 64 00
5007 108E01FF           LDY   #$01FF ;10 8E 01 FF
500B 863C               LDA   #60    ;86 60
500D 7F1902             CLR   6402   ;7F 64 02
5010 7F1903             CLR   6403   ;7F 64 03
5013 A780       CLEAR   STA   ,X+    ;A7 80
5015 313F               LEAY  -1,Y   ;31 3F
5017 26FA               BNE   CLEAR  ;26 FA
5019 869F               LDA   #$9F   ;86 9F
501B BE1900             LDX   6400   ;BE 64 00
501E A784               STA   ,X     ;A7 84
5020 BD1F46     KBD     JSR   8006   ;BD 80 06
5023 27FB               BEQ   KBD    ;27 FB
5025 815E               CMPA  #$5E   ;81 5E
5027 2605               BNE   DOWN   ;26 05
5029 BD5050             JSR   UP     ;BD 64 53
502C 20F2               BRA   KBD    ;20 F2
502E 810A       DOWN    CMPA  #$0A   ;81 0A
5030 2605               BNE   LEFT   ;26 05
5032 BD5051             JSR   MDWN   ;BD 64 54
5035 20E9               BRA   KBD    ;20 E9
5037 8108       LEFT    CMPA  #08    ;81 08
5039 2605               BNE   RIGHT  ;26 05
503B BD5052             JSR   MLFT   ;BD 64 55
503E 20E0               BRA   KBD    ;20 E0
5040 8109       RIGHT   CMPA  #09    ;81 09
5042 2605               BNE   BREAK  ;26 05
5044 BD5053             JSR   MRGHT  ;BD 64 56
5047 20D7               BRA   KBD    ;20 D7
5049 8103       BREAK   CMPA  #03    ;81 03
504B 2702               BEQ   END    ;27 02
504D 20D1               BRA   KBD    ;20 D1
504F 39         END     RTS          ;39
5050 39         UP      RTS          ;39
5051 39         MDWN    RTS          ;39
5052 39         MLFT    RTS          ;39
5053 39         MRGHT   RTS          ;39
```

This is because my assembler assumes a decimal value with digits 0-9 only unless it is told otherwise by preceding the value with a $. **Listing seven** was arrived at by assembling having inserted $ signs after the # in the operand columns of instruction lines with addresses $5007, $501B, $5025 and $502E. It is then obvious that the generated code (left hand side of **listing seven**) is quite different from the expected code in the book ('comment' column on the right hand side of last month's listing). Line at $5001, for instance, is 8E0190 rather than 8E0400; the get keypress JSR is BD1F46 rather than our familiar BD8006. Generated object code is always shown in hexadecimal. As the required generated values are identical to the operands of the original source listing, the source must be given in hexadecimal but not preceded by a $ as required by my (Dream) assembler. DECIMAL values have therefore been assumed excepting those earlier mentioned instances where the operand contained hex digits A-F. If you experienced the same problem the next stage is to insert $ as necessary throughout the source code where the value given in the operand column is identical to the required generated object code. The resulting assembly is still likely to contain discrepancies because the calls to the UP, MDWN, MLFT and MRGHT subroutines use the extended mode=actual address type of instruction, JSR and unless you assemble at the address the original is intended to run from, these addresses will be appropriate for the current assemble, as in **listing eight**. Apart from making the code position dependent, — that is, it must be run from the location assembled for — had it not been for other extended mode instructions such as

STX 6400
CLR 6402

the code would execute correctly despite being run from a different position. However, the above makes it an entirely different proposition that can be tackled in a

There is no listing nine. We do not know why — we merely observe this fact.

Listing 8

```
5001 8E0400     6404    LDX   #$0400 ;8E 04 00
5004 BF6400             STX   $6400  ;BF 64 00
5007 108E01FF           LDY   #$01FF ;10 8E 01
500B 8660               LDA   #$60   ;86 60
500D 7F6402             CLR   $6402  ;7F 64 02
5010 7F6403             CLR   $6403  ;7F 64 03
5013 A780       CLEAR   STA   ,X+    ;A7 80
5015 313F               LEAY  -1,Y   ;31 3F
5017 26FA               BNE   CLEAR  ;26 FA
5019 869F               LDA   #$9F   ;86 9F
501B BE6400             LDX   $6400  ;BE 64 00
501E A784               STA   ,X     ;A7 84
5020 BD8006     KBD     JSR   $8006  ;BD 80 06
5023 27FB               BEQ   KBD    ;27 FB
5025 815E               CMPA  #$5E   ;81 5E
5027 2605               BNE   DOWN   ;26 05
5029 BD5050             JSR   UP     ;BD 64 53
502C 20F2               BRA   KBD    ;20 F2
502E 810A       DOWN    CMPA  #$0A   ;81 0A
5030 2605               BNE   LEFT   ;26 05
5032 BD5051             JSR   MDWN   ;BD 64 54
5035 20E9               BRA   KBD    ;20 E9
5037 8108       LEFT    CMPA  #$08   ;81 08
5039 2605               BNE   RIGHT  ;26 05
503B BD5052             JSR   MLFT   ;BD 64 55
503E 20E0               BRA   KBD    ;20 E0
5040 8109       RIGHT   CMPA  #$09   ;81 09
5042 2605               BNE   BREAK  ;26 05
5044 BD5053             JSR   MRGHT  ;BD 64 56
5047 20D7               BRA   KBD    ;20 D7
5049 8103       BREAK   CMPA  #$03   ;81 03
504B 2702               BEQ   END    ;27 02
504D 20D1               BRA   KBD    ;20 D1
504F 39         END     RTS          ;39
5050 39         UP      RTS          ;39
5051 39         MDWN    RTS          ;39
5052 39         MLFT    RTS          ;39
5053 39         MRGHT   RTS          ;39
```

number of different ways — see if you can understand at least one of them!

As locations 6400-6403 are used directly in operands and the value 6404 appears in the label column of the first instruction, one can assume that the code is intended to reside from $6400 in memory, with $6404 being the address of the first program instruction. Using Dream and assemblers with similar 'redirection' facilities, listing 8 can be instantly adapted to match the original generated code by including the lines

ORG $6404
PUT $5001

before the first program line. The listing of this is not included as I am continually exceeding my monthly space allowance! The generated code is then identical to the book and occupies $6404-$6456 inclusive. This can be saved using

CSAVEM"BLOB5001",&H5001,&H5053,&H5001

and reloaded for use with

CLOADM'BLOB5001',&H1403

(resaved from there with

CSAVEM"BLOB6404",&H6404,&H6456,&H6404

if wished). Remember that the program also uses locations $6400-$6403 as variables space, so a clear of at least CLEAR-200,&H6400 is required before running the code.

A second option with the code should you not be able to generate it to execute it from $6404 would be to leave it position dependent but to change the addresses of the variables space from $6400-$6403 to be within our chunk of code. This involves creating variables space using label names and an assembler directive that may be slightly different on your assembler but is RMB (Reserve Memory Bytes) in Dream. One can usually reserve any number of bytes using an RMB directive. To avoid having to totally understand the program logic when trying to adapt similar published listings to run on your systems, I suggest that you adopt the method of allocating a label name per possible variables address and substituting actual addresses in source code operands with its respective label name. That is, in this double byte units, sometimes as double byte units. Rather than have to sort out each case individually, give each byte a label then substitute each address with its label, which should result in a position dependent but working program as in listing 10. The code is saved using

CSAVEM"YBLOB10",&H5001,&H5057,&H5005

and is loaded/executable from that address only. N.B. the use of the third parameter in the CSAVEM — the default EXEC address which is not the first byte saved if including the variables data area $5001-$5004 in the saved chunk of code (my preference so that you don't forget about such areas when fitting several chunks of machine code close together in memory!).

Another option is to make the entire chunk of code relocatable so that if will run from wherever we wish to load it at any one time. To achieve this is a relatively simple task if any actual memory addresses have been substituted with label names as in **listing 10**.

Extended address instructions need to be amended to relocatable types. JSRs, as we have met previously, should be replaced with BSRs (Branch to Subroutine). To make references to variables relocatable, if they have been given label names as in **listing 10**, simply append, PCR (Program Counter Relative) to the operand — or your assembler's equivalent as this, again, may be an area that differs slightly and perhaps, PC rather than, PCR is needed (Dream actually works with either). As with branch instructions that generate code to branch "Relative to the end of the current program instruction". Program counter relative access of variables causes code to be generated that refers to

Listing 10

```
5001                WORK0   RMB    1        ;FOR 6400
5002                WORK1   RMB    1        ;    6401
5003                WORK2   RMB    1        ;    6402
5004                WORK3   RMB    1        ;    6403
5005
5005 8E0400         6404    LDX    #$0400
5008 BF5001                 STX    WORK0
500B 108E01FF               LDY    #$01FF
500F 8660                   LDA    #$60
5011 7F5003                 CLR    WORK2
5014 7F5004                 CLR    WORK3
5017 A780           CLEAR   STA    ,X+
5019 313F                   LEAY   -1,Y
501B 26FA                   BNE    CLEAR
501D 869F                   LDA    #$9F
501F BE5001                 LDX    WORK0
5022 A784                   STA    ,X
5024 BD8006         KBD     JSR    $8006
5027 27FB                   BEQ    KBD
5029 815E                   CMPA   #$5E
502B 2605                   BNE    DOWN
502D BD5054                 JSR    UP
5030 20F2                   BRA    KBD
5032 810A           DOWN    CMPA   #$0A
5034 2605                   BNE    LEFT
5036 BD5055                 JSR    MDWN
5039 20E9                   BRA    KBD
503B 8108           LEFT    CMPA   #$08
503D 2605                   BNE    RIGHT
503F BD5056                 JSR    MLFT
5042 20E0                   BRA    KBD
5044 8109           RIGHT   CMPA   #$09
5046 2605                   BNE    BREAK
5048 BD5057                 JSR    MRGHT
504B 20D7                   BRA    KBD
504D 8103           BREAK   CMPA   #$03
504F 2702                   BEQ    END
5051 20D1                   BRA    KBD
5053 39             END     RTS
5054 39             UP      RTS
5055 39             MDWN    RTS
5056 39             MLFT    RTS
5057 39             MRGHT   RTS
```

Listing 11

```
5001 160004                 LBRA   GO
5004
5004                WORK0   RMB    1        ;FOR 6400
5005                WORK1   RMB    1        ;    6401
5006                WORK2   RMB    1        ;    6402
5007                WORK3   RMB    1        ;    6403
5008
5008 8E0400         GO      LDX    #$0400
500B AF8CF6                 STX    WORK0,PCR
500E 108E01FF               LDY    #$01FF
5012 8660                   LDA    #$60
5014 6F8CEF                 CLR    WORK2,PCR
5017 6F8CED                 CLR    WORK3,PCR
501A A780           CLEAR   STA    ,X+
501C 313F                   LEAY   -1,Y
501E 26FA                   BNE    CLEAR
5020 869F                   LDA    #$9F
5022 AE8CDF                 LDX    WORK0,PCR
5025 A784                   STA    ,X
5027 BD8006         KBD     JSR    $8006
502A 27FB                   BEQ    KBD
502C 815E                   CMPA   #$5E
502E 2604                   BNE    DOWN
5030 8D21                   BSR    UP
5032 20F3                   BRA    KBD
5034 810A           DOWN    CMPA   #$0A
5036 2604                   BNE    LEFT
5038 8D1A                   BSR    MDWN
503A 20EB                   BRA    KBD
503C 8108           LEFT    CMPA   #$08
503E 2604                   BNE    RIGHT
5040 8D13                   BSR    MLFT
5042 20E3                   BRA    KBD
5044 8109           RIGHT   CMPA   #$09
5046 2604                   BNE    BREAK
5048 8D0C                   BSR    MRGHT
504A 20DB                   BRA    KBD
504C 8103           BREAK   CMPA   #$03
504E 2702                   BEQ    END
5050 20D5                   BRA    KBD
5052 39             END     RTS
5053 39             UP      RTS
5054 39             MDWN    RTS
5055 39             MLFT    RTS
5056 39             MRGHT   RTS
```

```
Listing 12
UP      LDA     #00      ;86 00
        CMPA    6403     ;B1 64 03
        BEQ     ENDUP    ;27 0E
        LDA     #60      ;86 60
        STA     ,X       ;A7 84
        DEC     6403     ;7A 64 03
        LEAX    -32,X    ;30 88 E0
        LDA     #9F      ;86 9F
        STA     ,X       ;A7 84
ENDUP   RTS              ;39

MDWN    LDA     #15      ;86 0F
        CMPA    6403     ;B1 64 03
        BEQ     ENDDWN   ;27 0E
        LDA     #60      ;86 60
        STA     ,X       ;A7 84
        INC     6403     ;7C 64 03
        LEAX    32,X     ;30 88 20
        LDA     #9F      ;86 9F
        STA     ,X       ;A7 84
ENDDWN  RTS              ;39

MLFT    LDA     #00      ;86 00
        CMPA    6402     ;B1 64 02
        BEQ     ENDLFT   ;27 0D
        LDA     #60      ;86 60
        STA     ,X       ;A7 84
        DEC     6402     ;74 64 02
        LEAX    -1,X     ;30 1F
        LDA     #9F      ;86 9F
        STA     ,X       ;A7 84
ENDLFT  RTS              ;39

MRGHT   LDA     #1F      ;86 1F
        CMPA    6402     ;B1 64 02
        BEQ     ENDRGT   ;27 0D
        LDA     #60      ;86 60
        STA     ,X       ;A7 84
        INC     6402     ;7C 64 02
        LEAX    1,X      ;30 01
        LDA     #9F      ;86 9F
        STA     ,X       ;A7 84
ENDRGT  RTS              ;39
```

the memory location 'relative to' — that is, a fixed distance from, "the end of the current program instructions". The distance value is calculated for the program as it stands when assembled. If instructions are added or deleted between the variable's definition (the source line that gives it a label) and program instructions that refer to the label ,PCR, the reassemble automatically calculates and generates the new distance value as part of the generated object code for that instruction. The length of data generated for instructions varies according to the instruction mnemonic and operand. Unlike the standard (short) and long branch instructions, one doesn't need to differentiate regarding the distance in ,PCR type instructions. The assembler generates a 'distance' value in one byte for a variable within a range of +127/-128 bytes from the start address of the next program byte, as within this example, or two bytes for the +32767/-32768 range.

One final technique that I have included in the program is that I prefer the variables area to be seen to be part of the program and for the EXEC address to be the same as the load address to overcome memory lapses — a condition often encountered when working under pressure. If using my own machine code routines from Basic programs, I also like to have a variables area at a fixed place near the beginning of a machine code program so that values (parameters) can be PEEKed/POKEd easily between the Basic and machine code. I therefore like to start off machine code programs with a LBRA (Long Branch) to the real beginning of the program followed by a variables area. I use a long branch even where a short one would do so that I always know, regardless of the length of the variables area, that my Basic/machine code parameter areas will always start at a fixed point — program load/exec address+3 rather than have some program parameter areas commencing at +2 (the length of a short branch instruction) and others a +3, three being the length of a long branch instruction. Although Dream has not thrown it out (probably because nothing attempts to use it), Dream's label names should commence with an alphabetic character rather than the 6404 appearing at the beginning of the original program. I have therefore replaced this with the label name GO for **listing 11**.

Monthly workout

I suspect that, with the number of listings this month, it is touch and go as to whether this final listing makes it into print — but I'll chance it! **Listing 12** is the cursor movement routines as they appear in the book. Try incorporating them, in relocatable form, in **listing 11** (or your equivalent).

Crossword

The fourth Dragon Crossword is here to perplex you with perfidious posers from the Dragon's publishing history. And we have the results from crossword two: the fortunate few, picked out from the hat, were Paul Harrison of Brosely, whose entry arrived on the back of what looks like an electronic circuit for a single LED, and Jack Lund, who wants a surprise. Surprises, we are good at.

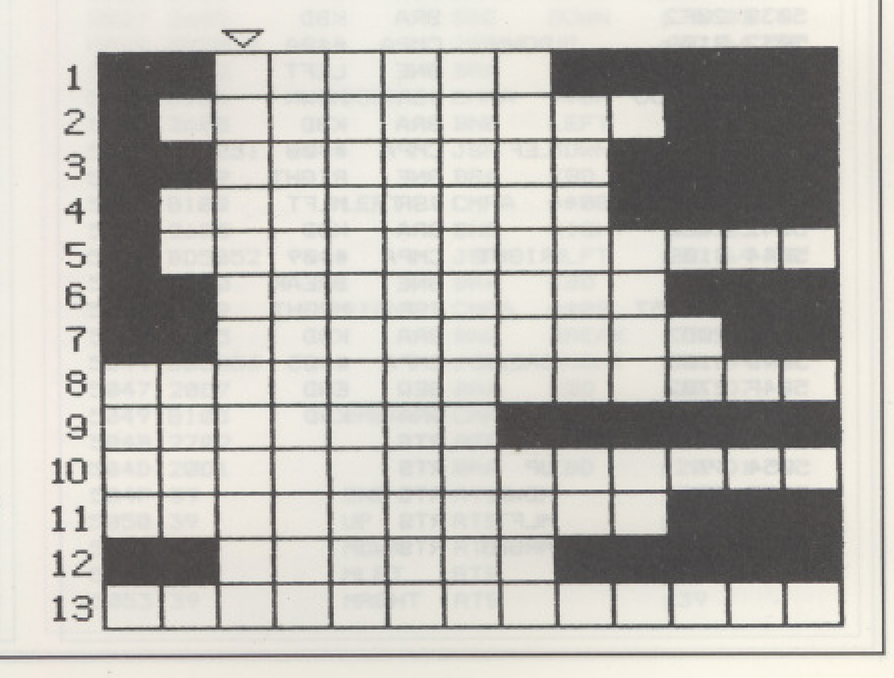
There will be a couple of free tapes from the Editor's Magic Bottomless Box for the first correct entries to reach us each month. You can even try telling us which tapes you'd like in an ideal world. It all depends on what we can find.

And you don't have to cut up your *Dragon User*, either — entries can be written out on a photostat or a plain piece of paper, as long as we can read them. And as long as there are no official secrets on the back!



All this month's answers are names of Dragon software. When the crossword is complete, the column marked with an arrow will spell out a phrase.

1. It's very hard to say — almost sounds like a pasty man! (6)
2. Mix mud on meal to build a person up (6,3)
3. Croak, usually goes with a leap! (7)
4. A learner they tax for sport (8)
5. Avast! brews us chalk mixture to raise the Jolly Roger! (12)
6. Birdman lived here (8)
7. Sounds like he only steals red stones! (4,5)
8. Three stories of glory in teeth kit (3,3,7)
9. Hold your lemons — the Mexican robber has nothing on! (7)
10. Does he kill ghosts? (7,6)
11. Team up with 9 to steal hours! (4,6)
12. Get away to do this (6)
13. Nothing in circus crimes to take you to the stars (6,7)



Total Profits

*Alpha Centauri School of Cost Effective Plundering graduate **Julian Brown** has some tips for traders.*

EVEN after all this time many of you out there are still struggling, trying to make a living out of the much simplified trading, while people like me have made multiple visits to the second universe. So out of the kindness of my empty bank-account I have decided that all you sufferers out there need my help badly. (*Anyone who makes a living writing for Dragon User needs help, mate. — Ed.*)

I have been limited on space, so this will have to be brief, so listen good!

Here are the basic tips:

1. Avoid anarchy worlds, unless you want to be used for target practice.
2. Keep your ship in a good state of repair at all times.
3. Never pay the pirates.
4. Know where the repair centres are (see **table one**).
5. Make sure you have a good trading route.

Table one

PLANET	COORDINATES	PLANET	COORDINATES
JJAMON	1:3:5	ZEEISS	1:5:3
ISPKKA	2:4:2	ZEEFEE	2:3:3
BASYYA	2:6:4	VAXURY	2:3:5
THOKKI	4:1:1	MENURI	4:3:3
SONSKA	4:5:3	PLAORA	5:3:2
SKAFFA (anarchy)	5:5:2	VRBONE	5:1:4
SHOTTE	5:2:5	IANTHE	5:5:5
PCHION	6:1:3	JOWICE	6:2:6
JOWEXA	7:4:3	LLAKKA	8:6:1
WRISCH	8:3:4	STUORI	8:2:6
TUSSPO	8:6:6	YYASPI	10:1:2
YYAASH	10:3:5		

The format for the co-ordinates is sector:x:y. All of the above have repair facilities but some are cheaper than others; Skaffa is the most expensive planet and is best avoided at all times.

The recommended trade route is between Tasvee (7:1:6) and Stuori (8:2:6). Buy uranium at Tasvee and sell it at Stuori but don't buy anything for the return journey.

Technicalities

Taxes: planets that have landing taxes are best avoided unless you know what you are doing. If you do visit them make sure it's a brief visit.

At the end of each combat you automatically collect cargo left by the pirates. This cargo is always a collection of three items: gems, hi-techs and narcotics. With no load and one cargo hold you collect two items; every hold you buy increases this number to a maximum of six; after this the number reverts to zero and then two again.

Until you start to collect six or more items you will find no narcotics; ethics aside, this is more of a benefit than it sounds. Trying to get narcotics is nearly impossible. Tailoring the amount you collect is easy and best illustrated with by **table two**.

Table two

Number of holds	Maximum no. of cargo units
1	2
2	4
3	6
4	0
5	2

When you have the maximum number for your hold, regardless of whether it is food, textiles, etc., you can't collect any more. But by adding other cargo so the total in the hold is more than the maximum listed above, you will receive extra.

For example, if you had one hold with a single unit of food, you would collect 1 unit of booty, but if you had instead 3 units of food you would collect 7 units of booty.

It is best to tailor the results so that you recieve 5 units of booty, which is the maximum safe number.

For example, if you have 1 hold, carry 5 units of cargo, if you have two holds carry either 15 or 9 units of cargo, with three holds carry 1, 11, 19 or 25 units of cargo.

Don't forget: always pay import taxes and avoid narcotics.

Upgrades

If you take my advice and start trading between Tasvee and Stuori then upgrades will be readily available to you. Don't be tempted to buy extras for your ship as soon as you have enough money, wait till you have about twice the amount of money you need, in case of disaster.

Always expand your cargo hold first, which increases the cash you gain per trip if you use a full cargo hold. Follow this up with fuel tanks, then lasers, then computer expansions.

Once you have all of these buy the drilling kit, eight wellcaps and a hyperdrive. With all this done, you are on your own.

Drilling

Find an asteroid with a reserve of at least 10 and find its weakest point in the crust. Using a NEW drill, start drilling, stopping every time the drill bit reaches about 40 degrees, and allow it to cool. Once the asterbuggy is full then cap the well.

Go to a space station (you won't find one in the same galaxy) and sell your load. Once this is done, it's worth waiting a few lifetimes while the value of asteroil goes back up again. And get yourself a new drill.

The Prophet

Once you have collected cash in the region of a few billion credits you should start to receive telexes from the prophet; fly to Phaare and you will receive a shard. All that remains is to fly down the nearest black hole.

Universe two is a nasty place. Everything costs ten times as much, there are dead ships to be salvaged, and worst of all the save games don't work because of a bug in the program. Last but not least, even though the names of the cargo items don't change, you'll have to find new initials for them, except the luxuries. The names are supposed to change, but another bug in the program prevents it.

Finally, if you do get stuck and need continual help I can supply a completed version of my editor program that runs on ALL Dragons for £1 plus 75p for postage and packing. Just send your request to: Julian Brown, 2 Lavendar Cottages, Salmon's Lane, Prestwood, Bucks HP16 0PY. Don't forget to include your own address, and don't forget the money, or you may not receive anything!

READPC

Martin Vermeer devises a program to read PC discs on a Dragon

A program to read DragonDOS diskettes on a PC has been available from Compusense for some time now. One of my friends who owns both a Dragon and PC is very happy with it: it is well made and easy to use. Personally however I have been much more interested in software to do the reverse: transfer PC software in Basic, Pascal and C to run on my little Dragon.

The problem

More than a year ago I was faced with a professional need to get a number of large files from a PC diskette into my Dragon. The solution I chose was to purchase the PC-READ software from D.P. Johnson — an American professional OS-9 programmer — which is made for the Color Computer and also runs on the Canadian-made Sardis ST-2900. The latter machine is the one used by a friend of mine — he built it himself from the circuit boards — and I asked him to transfer my files to Color Computer OS-9 format diskettes, which the Dragon can read.

A little complicated, though. I suppose there must be many Dragonists who would like to use some of the supposedly abundant PC software on their home machine, or who use a PC at work and have made programs and texts for it which they would like to use at home also. There might be even frustrated PC owners fascinated by the idea of upgrading to the Dragon's superior operating system, but held back by the difficulty of transferring their materials. Anyway, those days are over now, as the following programs allow for the easy and quick transfer of even large Ascii files — texts, software source codes, whatever — from MS-DOS diskettes to either DragonDOS or OS-9.

Using READPC

Accompanying this text is a Dragon Basic listing of a program which will read an MS-DOS diskette, present you with a directory and transfer the file chosen by you by number to ... a cassette tape, as a Dragon Basic Ascii file. This unconventional solution was chosen (1) to allow single-drive owners to use it and (2) not to place any limit on transferable file size.

LOAD the program and RUN it; wait until the MS-DOS directory appears on the screen. Choose the file to be transferred by number and press ENTER. Make sure the tape recorder is connected and ready, and an empty cassette lined up and ready to record, and both Play and Record buttons down. Especially with large files, the transfer may take some time, so be patient.

In case the directory is larger than the screen, you can stop the listing by pressing the shifted AT key. You can repeat the directory listing by just pressing ENTER, ie by specifying file number 0. A limitation of this program is that it only 'sees' files in the MS-DOS root directory.

Be careful if you are transferring text files in a non-British European language: the 'extra' characters which occur in Spanish, German and Swedish/Finnish are treated by MS-DOS in a way which violates the Ascii standard. Where Ascii has those characters in the same places as the British or US Ascii set has square brackets, braces and the like, IBM has been creative in using 8-bit codes instead. All 8-bit codes are replaced by READPC with question marks. So are braces and other characters that the Dragon's cassette I/O system does not recognise. In case you want these transferred — important in for example C source code — you should either replace them on your PC, or be prepared to correct them afterwards on your Dragon.

The above program puts your file to cassette as an Ascii file, which can be read by word processors like *Telewriter*. Alternatively, a Basic program can access the file as a data file by using the well known LINE INPUT $-1 statement, which reads exactly one line of text (from carriage return to carriage return) from the file into a single text string.

If your text happens to be a Microsoft Basic program, you can try to load it straight away into the memory and edit it by the Dragon's own line editor. It is likely however that it needs some cleaning up before this works.

Getting to OS-9

To transfer the file from tape further to OS-9 requires additional software written to transfer Ascii tape files in general to OS-9. Note the building block approach: the OS-9 philosophy is addictive! What I did was simply write an OS-9 device driver for the Dragon's cassette recorder; it treats the device as a sequential character file which is only capable of input. With the driver goes a descriptor, a little table of values defining the settings of the peripheral device. I obtained both by disassembling and suitably patching the ACIA51 serial port driver and its /T1 descriptor.

Perhaps these programs could be streamlined a little more, but they have been tested and work as they are. In order to use them, assemble these source codes and let them reside in your execution directory. Ready your cassette, BOOT OS-9 (always before every transfer) and load the modules CASS, CR, and COPY or LIST. To get your file on the screen, you would write:

OS9:list /cr (ENTER),

but presumably you are more interested in file transfer; this works with COPY as follows:

OS9:copy /cr pathname (ENTER)

Cleaning up

When listing the file received, you will generally notice that it contains besides carriage returns (Ascii 13) also line feeds (Ascii 10). This is typical for MS-DOS Ascii files, but the Dragon wants only carriage returns. A second probem is that the casette I/O system has generated spurious carriage ruturns in the file.

Both problems are corrected by the following little program 'f', which is written in Basic09. Write it in, SAVE it, PACK it and use it together with RunB, as follows:

OS9:f <oldfile >newfile (ENTER),

which will produce a new file with the proper amount of carriage returns. At the end of the file there will still be some junk however, but that is easily edited out. READPC does not check for the precise file length of an MS-DOS file.

Nonstandard formats

I have succeeded in reading at least the 720K 3.5)in MS-DOS diskettes used in many portable machines by the following modifications: 1) change the parameter DSTART in line 50 to 7 2) change line 130 to:

130 FOR LSN=FS TO FS+2

For other formats, other parameters should be changed (number of sides, number of sectors per track), but I have no experience with this. There are plenty of good books on MS-DOS disc structure (in public libraries, for example) to help you, as well as the excellent (and extensively stolen) Norton's Utilities.

You could also cannibalise a general-purpose disc browsing utility from the READPC sector read routine. When doing so, be aware that this routine is literally general-purpose, and not limited to sector sizes of 256 bytes, but reads any sector size. At the moment, the sector read is double density only, but even this limitation can be removed: the Dragon hardware supports single density disc I/O and only waits for the programmer to use it!

Listing 1

```
LISTING 1:  READPC for DragonDOS.

10 PRINT"wait...":PCLEAR1:CLEAR4000,29900:GOSUB1050
20 REM nsec SECTORS/TRACK, fstart FILE ALLOC TABLE START
30 REM dstart DIRECTORY START, ksides NO. OF SIDES
40 REM csize CLUSTER SIZE, cl FIRST CLUSTER SECTOR NO.
50 NSEC=9:FSTART=1:DSTART=5:KSIDES=2:CSIZE=2:C1=12
60 REM main program of readpc
70 S1$="":S2$="":S3$="":S$="12345678123"
80 DIM DR$(112):FOR I=1 TO 112:DR$(I)=STRING$(11,0)
   :NEXT I
90 DIM AT(112),BY(112),FP(112)
100 CLS:PRINT´110,"readpc":PRINT:
    PRINTTAB(8)"READS pc DISKETTES":
    PRINTTAB(10)"ON THE dragon":PRINT:
    PRINTTAB(3)"(c) 1987 BY martin vermeer"
110 PRINT:PRINT:INPUT"   drive:";DV
120 FE=1
130 FOR LSN=FS TO FS+1
140 GOSUB640:GOSUB290:GOSUB710
150 NEXT LSN
160 DE=1
170 FOR LSN=DS TO C1-1
180 GOSUB 640:GOSUB290:GOSUB500
190 NEXT LSN
200 GOSUB210:GOSUB770:GOSUB200
210 REM print directory
220 PRINT
230 FOR I=1 TO DE-1
240 A$=LEFT$(DR$(I),1):IF A$>="Ä"OR A$<="´" THEN 270
250 PRINT I;DR$(I);
260 IF NOT(AT(I)AND&H1E)THEN
    PRINTUSING"§§§§§§ §§§§"; BY(I);FP(I) ELSE PRINT " *** "
270 NEXT I
280 RETURN
290 REM disk sector read routine
300 REM for ms-dos disk format
310 BUFFER=30100:ADDR=30003
320 POKE&HFF48,&H04+DV-1:'START MOTOR,SEL DRIVE
330 POKE ADDR-1,&H24+DV-1:'SETUP DRIVE VAR
340 FOR I=1 TO 2:POKE&HFF40,&H57
350 IF PEEK(&HFF40) AND 1 THEN 350
360 NEXT I
370 POKE&HFF40,3:'RESTORE
380 IF PEEK(&HFF40) AND 1 THEN 380
390 IF TRK=0 THEN 430
400 FOR I=1 TO TRK:POKE&HFF40,&H57
410 IF PEEK(&HFF40) AND 1 THEN 410
420 NEXT I
430 POKE ADDR-2,SIDE
440 POKE&HFF42,SEC
450 EXEC ADDR:'SECTOR READ ROUTINE
460 REM sector read statistics:
470 NBYT=PEEK(BUF-2)*256+PEEK(BUF-1)-BUF
480 PRINT NBYT;"BYTES FOUND"
490 RETURN
500 REM save directory entries
510 PRINT"GETTING dir..."
520 FOR I=1 TO 16
530 AT(DE)=PEEK(BUF+11+32*(I-1))
540 BA=BUF+28+32*(I-1)
550 BY(DE)=PEEK(BA)+256*(PEEK(BA+1)+256*PEEK(BA+2))
560 BA=BUF+26+32*(I-1)
570 FP(DE)=PEEK(BA)+256*PEEK(BA+1)
580 BA=BUF+32*(I-1)
590 SP=VARPTR(S$)
600 POKE SP+2,INT(BA/256):POKE SP+3,BA AND 255:POKE SP,11
610 MID$(DR$(DE),1)=S$
620 DE=DE+1:NEXT I
630 RETURN
640 REM lsn TO tRK,sEC,hEAD conversion
650 REM lsn = lOGICAL sECTOR nUMBER
660 TRK=INT(LSN/(NSEC*KSIDES))
670 SEC=LSN-NSEC*KSIDES*TRK+1
680 IF SEC>NSEC THEN SIDE=&H8A:SEC=SEC-NSEC ELSE
    SIDE=&H88
690 PRINT:PRINT "lsn";LSN;"t";TRK;
    "s";SEC;"h";(SIDE-&H88)/2;"=========="
700 RETURN
710 REM get FAT to high mem:
720 PRINT"GETTING fat..."
730 FOR I=0 TO 511
740 POKE BUF+I+512*FE,PEEK(BUF+I)
750 NEXT I:FE=FE+1
760 RETURN
770 REM get file clusters & print
780 PRINT:INPUT"file NUMBER:";FI
790 IF FI<=0 THEN RETURN
800 PRINT:INPUT"OUTPUT device (0=SCRN, 1=CASS,
    2=PRINTER)";OD:
    IF OD<0 OR OD>2 THEN BEEP:GOTO800
810 IF OD=1 THEN OPEN"O",§-1,"PC-FILE"
820 PT=FP(FI):FS=BY(FI):GOSUB940
830 REM get cluster
840 LSN=C1+CS*(PT-2):GOSUB640:GOSUB290
850 GOSUB990
860 LSN=C1+CS*(PT-2)+1:GOSUB640:GOSUB290
870 GOSUB 990
880 PA=BUF+512+1.5*PT
890 IF PA=INT(PA) THEN PT=PEEK(PA)+(PEEK(PA+1)
    AND 15)*256
    ELSE PT=INT(PEEK(PA-0.5)/16) + PEEK(PA+0.5)*16
900 FS=FS-1024
910 IF FS>0 THEN 840
920 CLOSE§-1
930 RETURN
940 REM point to sector
950 SP=VARPTR(S1$):POKE SP+2,INT(BUF/256):
    POKE SP+3,BUF AND 255:POKE SP,200
960 SP=VARPTR(S2$):POKE SP+2,INT((BUF+200)/256):
    POKE SP+3, (BUF+200) AND 255:POKE SP,200
970 SP=VARPTR(S3$):POKE SP+2,INT((BUF+400)/256):
    POKE SP+3, (BUF+400) AND 255:POKE SP,112
980 RETURN
990 REM print sector
1000 PRINT§-OD,S1$;S2$;S3$;:RETURN
1050 FOR I=29901 TO 29914:READ X:POKE I,X:NEXT I:EXEC29901
1060 DATA 48,140,4,191,1,104,57,129,122,35,2,134,63,57
1100 FOR I=1 TO 100:READ X: POKE 30000+I,X:NEXT I: RETURN
1110 DATA 130,36,52,63,26,80,190,1,10,175
1120 DATA 140,83,48,140,54,191,1,10,134,255
1130 DATA 31,139,16,142,255,255,48,140,70,150
1140 DATA 35,138,1,151,35,150,34,230,140,216
1150 DATA 215,64,166,140,212,151,72,150,35,43
1160 DATA 10,49,63,38,248,32,14,214,35,42
1170 DATA 252,150,67,214,34,167,128,32,244,50
1180 DATA 108,214,34,15,72,175,140,19,174,140
1190 DATA 14,175,141,198,28,53,63,57,198,23
1200 DATA 53,63,57,57,80,198,247,58,252,255
```

Listing 2

```
*
* LISTING 2: Cass Dragon cassette read driver
*
          mod    cassend,cassnam,$E1,$81,cassent,cassmem
*
ADconv    equ       $ff20
motor     equ       $ff21
*
pwc       rmb       $1     ; pulse width counter
pointer   rmb       $1
length    rmb       $1
bitcnt    rmb       $1
bpf       rmb       $1     ; bit phase flag
buffer    rmb       $0120
cassmem   equ       .
*
          FCB       $03
cassnam   FCs       "Cass"
          FCB       $05
*
cassent   LBRA      init
          LBRA      read
          LBRA      write
          LBRA      getstat
          LBRA      setstat
          LBRA      term
*
getAD     inc       pwc,U
          ldb       ADconv
          rorb
          rts
*
getwave   clr       pwc,U
          tst       bpf,U
          bne       phset
```

```
phclr    bsr     tic
tac      bsr     getAD
         bcc     tac
         rts
phset    bsr     tac
tic      bsr     getAD
         bcs     tic
         rts
*
bitin    bsr     getwave
         ldb     pwc,U
         decb
         cmpb    §$12
         rts
*
bytin    lda     §$8
         sta     bitcnt,U
nextbit  bsr     bitin
         rora
         dec     bitcnt,U
         bne     nextbit
         rts
*
bump     clr     pwc,U
         bsr     tic
         bra     sl
pit      clr     pwc,U
         bsr     tac
sl       ldb     pwc,U
         cmpb    §$12
         bhi     s2
         cmpb    §$0a
         rts
s2       clr     bitcnt,U
         rts
*
cason    lda     motor
         ora     §$8
         sta     motor
         ldx     §$8000
delay    leax    -1,x
         bne     delay
         rts
*
csrdon   bsr     cason
         clr     bitcnt,U  ; state machine:
                                reads bits 010101
loop     bsr     phclr       ; el <0- 0+> e2
el       bsr     bump        ; !         !
         bhi     e3          ; 1+        1-
e2       bsr     pit         ; V         V
         blo     e4          ; e3 <1+ 1-> e4
         inc     bitcnt,U    ; !         !
         lda     bitcnt,U    ; 0-        0+
         cmpa    §$60        ; V         V
         bra     out         ; dec       inc
e3       bsr     pit         ; $60       $00
         bhi     el
e4       bsr     bump
         blo     e2
         dec     bitcnt,U
         lda     bitcnt,U
         adda    §$60
out      bne     loop
         sta     bpf,U
*
         clra
again    lbsr    bitin
         rora                ; accumulate bits
         cmpa    §$3c        ; look for sync byte
         bne     again
         rts
*
casoff   lda     motor
         anda    §$f7
         sta     motor
         rts
*
```

```
write    comb
         ldb     §245      ; write error
         rts
*
getstat  clrb
         rts
*
setstat  equ     getstat
term     equ     getstat
*
init     orcc    §$50
         lbsr    csrdon    ; start cassette
         lbsr    bytin
         cmpa    §$00
         bne     err246    ; not ready error
         lbsr    bytin
         pshs    a
nextbyt  lbsr    bytin
         dec     ,s
         bne     nextbyt
         puls    a
         lbsr    bytin
         lbsr    bytin
         clr     pointer,U
         clr     length,U
         bsr     casoff
         andcc   §$af
         rts
*
err246   bsr     casoff
         comb                ; set C bit
         ldb     §246
         andcc   §$af
         rts
*
read     orcc    §$50
         ldb     pointer,U
         bne     getchar
         lbsr    csrdon    ; get block first
         lbsr    bytin
         cmpa    §$ff
         beq     err211    ; end of file
         lbsr    bytin
         sta     length,U
         pshs    a
         leax    buffer+$10,U
nextb    lbsr    bytin
         sta     ,X+         ; store into buffer
         dec     ,S
         bne     nextb
         puls    a
         lbsr    bytin
         lbsr    bytin
         lbsr    casoff
         clrb
*
getchar  leax    buffer+$10,U
         abx
         lda     ,X
         incb
         cmpb    length,U
         blo     over
         clr     pointer,U
         bra     exit
over     inc     pointer,U
exit     clrb
         andcc   §$af
         rts
*
err211   lbsr    casoff
         comb
         ldb     §211
         andcc   §$af
         rts
*
         emod
cassend  EQU     *
         END
```

```
*
* LISTING 3: /CR Dragon cassette reader device
                                 descriptor
*
         mod      crend,crnam,$f1,$81,crent,crmem
*
         rmb      $31
crmem    equ      .
*
         FCB      $03
         FCB      $FF
         FCB      $FF
         FCB      $04
         FCB      $1A
         FCB      $00
         FCB      $00
         FCB      $01
         FCB      $00
         FCB      $00
         FCB      $00
         FCB      $00
         FCB      $00
         FCB      $18
         FCB      $08
         FCB      $18
         FCB      $0D
         FCB      $1B
         FCB      $04
         FCB      $01
         FCB      $17
         FCB      $03
         FCB      $05
         FCB      $08
         FCB      $07
         FCB      $60
         FCB      $1E
         FCB      $00
         FCB      $2C
         FCB      $00
         FCB      $00
*
crnam    fcs      "CR"     descriptor name
         fcs      "scf"    associated file manager
crent    FCs      "Cass"   associated driver name

         emod
crend    EQU      *
```

```
LISTING 4: Cleanup procedure (Basic09)

PROCEDURE f
 0000            DIM c:BYTE
 0007            WHILE NOT(EOF(§0)) DO
 0011              GET §0,c
 001A              IF c>31 AND c<128 THEN PUT §1,c
 0035              ENDIF
 0037              IF c=$0A THEN c=$0D Ö PUT §1,c
 0054              ENDIF
 0056            ENDWHILE

(where "Ö" stands for "backslash" and "§" for "hash")
```

How green is my black

***Paul Reid** gets bored with black on green, and invents an inversion.*

THE video display generator (vdg) chip in the Dragon, the one that dictates that we are stuck with a really rather bland black-on-green text display and black screen border, is the Motorola MC684YP, located on the main board near (on the Dragon) the cassette and joystick sockets.

Consulting the specification sheet for this particular chip, I discovered that pin 32 can be used to invert the screen colours in Alpha (text) mode. This is done by grounding the pin, since normal video is displayed with pin 32 'high' at +5 volts. As with most mods however, there is a trade-off: the normal inverse video characters, the lower case set, are not generated in this way and are unaffected by the mod. This has both advantages and disadvantages. On the plus side, a Basic program listing on the screen which includes lower case text intended for output to a printer, will appear as 'normal' text - but inverted along with everything else, rather than as blocks of inverted characters which if they have upper case characters mixed in with them, makes the whole thing very difficult to read. On the minus side, it is difficult at times when inputting lower case text in the inverted state, to know whether you are in upper or lower case mode!

Not Ascii

A second trade-off is that because the cursor is not an Alpha ascii character, it does not invert. Instead, it becomes a flashing black graphics block on a black background - somewhat difficult to see!

Nonetheless, these apparent problems can be overcome. The essence of this modification is that it is switchable - even during the running of a program, so that if confusion exists, simply by switching back to normal video, the display instantly returns to the original format. Moreover, the problem of the 'missing' cursor can be overcome by replacing the graphics block with a hyphen or other Alpha character using the routine published in DU a while ago. (DU - how about reprinting the item referred to?) (*At the time of asking I still haven't managed to identify it - Ed.*).

The advantages of using this method to achieve inverted display are considerable when it is considered how much Basic programming is required to achieve inverted spaces and punctuation marks in print statements!

Here then, are the instructions for carrying out the mod which requires only the (careful) use of a soldering iron, preferably low wattage with a miniature tip, a drill for making the hole in the case in which to fit the switch and the switch itself, a single pole double throw (spdt) type along with three lengths of insulated wire each about 40cm long. Switch off and disconnect the power supply.

I discovered that pin 32 can be used to invert the screen colons in Alpha (text) mode

(see **figure 1**). On some boards, the vdg is fitted in a socket, if this is so with yours, remove the chip taking not to bend the legs. An IC fitting/removing tool should be

After separating the two halves of the Dragon's case by removing the four screws located one in each corner of the base (well, there may be some Dragon owners who haven't been inside theirs yet!), drill a suitable sized hole in the most convenient, accessible place for the switch, remember that you will want to be able to operate it while using the computer. Identify pin number 32 of the vdg chip and also pin 20 used if possible. Bend pin 32 outwards so that it is at 90 degrees to the other pins and solder one end of the three lengths of wire onto it. The other end of the wire should be soldered to the centre (common) pin of the switch. Next, solder one end of the second piece of wire to the upper wide part of pin 20, taking care not to bend the pin or to get solder on the thin lower part. The other end of the wire should be soldered to either of the two remaining pins of the switch. The third length of wire must be soldered between the remaining pin on the switch and the socket which pin 32 normally fits into. If on your Dragon, like mine, the vdg chip is soldered directly to the board, pin 32 should either be de-soldered (using braid or a pump from the underside of the board) and carefully pulled back through the board, or cut with a pair of side-cutters as close to the board as possible.

If it was removed, the vdg chip can now be refitted, taking care that pin 32 does not make contact with the wire now soldered in its place. Fit the switch to the case in the drilled hole and reassemble the case. Reconnect the power supply, monitor/tv lead and switch on. Don't worry if the display is normal, operate the new switch and see what happens!

Ten minutes

The next invertion will in no way affect the computer's operation, or necessitate any program changes, but if like me you are tired of that black-on-green display, it makes a refreshing change! Even if you are not a wizard with the soldering iron, provided you are careful not to use too much solder, you should have no problems and the results are well worth the 10 minutes or so to do the modification and the cost, which is only a few pence.

(Although this mod has been tried and tested, as usual, the author and Dragon User can accept no responsibility for any problems arising from the modification. We also suggest that you don't attempt a modification, even a simple one, inside your computer unless you are experienced at construction.)

Expert's Arcade Arena

Write to 'The Expert' at Dragon User
12-13 Little Newport St
London WC2H 7PP.

HELLO and welcome. I'd like to start this month's column with a handy POKE which makes all games load four times as fast, operate with infinite lives and play that national anthem. Unfortunately, I can't. But I've got lots of other goodies for you instead, including the first of the answers to the questions which I set in January. Please keep your letters coming in, especially those entries in the software survey. At present, *Shocktrooper* and *Rommel's Revenge* are the hottest contenders for the all time best arcade game, and Ed Scio has received over 75% of the votes and looks like winning the title of best programmer. That is, as dear Esther would say, unless you know better!

And so to business. As promised, here's some more from the one and only James Bonfield. Mr. Goodmeadow provides us with a new POKE for *Pitfiend* from Pocket Money.

Pitfiend cheat

Load the game using program C (see *Dragon User* September 1987) coupled with (to change maze):

20 POKE7921,x:POKE7932,x

where x is a number 0-255 (default 3)

Unfortunately, I'm having a bit of trouble with this one. As James himself said, "It's a dodgy, experiment". And now for another grand unveiling from James, the final messages in *Copta Snatch* (but a good question is, 'did he play the game to get them?' Over to you, James.

"In *Copta Snatch*, when you land on the last platform, the screen displays:

RETURN TO HQ TO
COMPLETE MISSION

On returning, the final message is:

MISSION COMPLETED

He also says that while this is displayed, the computer plays a tune and 'springy things fly around the screen', but if you ask me it's only a little recognition for such a mammoth effort needed to complete the game!

James sent me a few maps, the *Escape* map being particularly interesting, but unfortunately, we weren't able to print it. I've listed below a few points to remember if you draw a map and would like us to consider printing it. No more *Copter Snatch* maps please!

1. Always use plain paper. If you need squares then slot a sheet of graph paper or squared paper underneath your plain and it will show through.
2. Always use a black pen, and preferably not something which blots, smudges, or goes patchy. Colours won't show up.
3. Try to ensure that lettering and details are really clear.
4. Maps should be large enough to be clear but preferably no larger than A4, and lettering should be of a readable size.

These technicalities should enable us to bring your efforts to the world if the opportunity arises.

And now for *Total Eclipse* fans (wake up Joe Brincat, are you recieving me?). Two loyal servants, Peter Macmillan and David Linsley, have replied to question 2 of January's pleas, and I am pleased to announce that Universe II does in fact exist. Peter informs me that it's a crazy world in which you trade in Chalk Dust, Blue Socks, Fog and Kangeroos. (Weird! 'nuff said.)

He also offers a helping hand to those who have Universe II and want to know how to play it — he thinks people are in such a situation, but I don't know why!

Thank you Peter and David. But the way, David, I'm not any of the people you suggested, especially Kenny Everett.

AND NOW THIS ... I must give you a *Wizard's Quest* POKE, so no-one else sends it in. I've had so many people send it to me that I've forgotten both their names (yawn). To give credit where credit is due, thank you Tom Wilkinson, James Bonfield, Michael Dunn and Sean Neale.

Wizard's Quest cheat

Load using program C, coupled with:

20 POKE 26494,33

This gives infinite lives.

And now, with most of the official business out of the way, you and I must have a serious talk. Sit down and pull your *Dragon User* up closer because I want a private word.

Now, I don't ask much of you folks, so you could at least grant me this one, tiny request. NO MORE CLEVER PEN NAMES PLEASE! Yes, you. I refer to you, of course, Dudley the Incredible Games Playing Goldfish. I don't want to be herring any more from you under this name as I cod very angry and discredit your carp pen name, so on your pike! For those interested, this fishy fellow, who sent me a most unusual letter, is in fact Simon Harrison, so now that you know his *real* name you'll never take him seriously again!!

Yet again I've reached the end of another page, so I must sign off. Keep your letters coming in, and lend your eyes next month for some help of *Catacomb Crisis*.

Communication

Problem: I am finding difficulty in obtaining floating joysticks.
Name: Scott Hadden
Address: 336, Cluny Place, Glenrothes, Fife KY7 4QY.

Problem: Need a printer cable for MCP-40 printer/plotter.
Name: Duncan Chambers
Address: 30 Park Avenue, South Shields, Tyne & Wear.

Problem: I need an Edit Plus cartridge for Dragon 32/64.
Name: W A Slater
Address: 44 Hope St., Brampton, Chesterfield, Derbyshire S40 1DG

Problem: Wanted: one Dragon controller for an unemployed person. Also wanted for a youth club — joysticks in any condition for the Dragon. Can repair if broken.
Name: Dennis Gates
Address: 194 Bek Road, Newton Hall Estate, Durham.

Problem: I am looking for an original copy of *Hotel on Mayfair* by Datacom. I will pay £4.00 for it.
Name: P. MacMillan
Address: Rose Cottage, Oakwood Road, Sling, Nr. Coleford, Glos. GL16 8JG.

Problem: Wanted: disc based word processor, *Telewriter* or *Tandy Scripsit*, suitable for Dragon 32. Write to me.
Name: Andy Fell
Address: 17 Gorse Farm Road, Thornhill Park, Nuneaton CV11 6TH.

Communication

Write down your problem on the coupon below (make it as brief and legible as possible) together with your name and addres and send it to Communication, 12/13 Little Newport Street, London WC2H 7PP.

Problem ....
....
....
....

Name ....
Address ....
....
....

Down in the dumps

Dragon User presents another screen dump

Seikosha GP100A

THIS basic program can dump the contents of the Dragon's hi-res screens 0, 2 and 4 directly to the Seikosha GP100A printer, in one of two sizes: mine (about 4in by 3in) and large (just under A4 size).

After the program has been typed in correctly and run, it is simple to use. Full instructions are in the program. You simply enter the PMODE which the screen in question was saved in, the size of the dump, and various other minor details, and the screen file, saved in the normal way (created by an art program, or from a game screen, etc.), is then loaded and dumped. It is worth pointing out that some utilities save screens by different methods. By one method, once a screen has been loaded it 'erases' any program in memory. The program allows for this: simply load in your screen (using CLOADM) beforehand, then the dumper program, and then chose option 2 from the menu.

This program is also available on tape from myself, priced £2.00 (UK). Cheques/POs payable to Andrew Hill, 13 Parry Jones Close, Blaina, Gwent NP3 3NH.

Notes: the Seikosha GP100A uses an unusual method of graphics printing. It reads 7 bits at a time instead of the more usual 8.

PMODE 1 and 3 colour screens can only be printed if you change the PMODE to 2 or 4 so that they are monochrome.

Andrew Hill

Dump C

```
10 REM *-------------------*
20 REM * - SCREEN DUMPER - *
30 REM *-------------------*
40 REM * [C] COPYRIGHT  87 *
50 REM *-------------------*
60 REM * WRITTEN BY A.HILL *
70 REM *-------------------*
80 CLEAR 500
90 CLS
100 PRINT"--------------------------------";
110 PRINT"  * * * s e i k o s h a * * *"
120 PRINT:PRINT  " * s c r e e n * d u m p e r *"
130 PRINT"--------------------------------";
140 PRINT" WELCOME TO 'seikosha screen        dumper', A PROGRAM WHICH ALLOWS YOU T
O 'DUMP' THE CONTENTS OF    YOUR HI-RES SCREEN (0/2/4) TO    THE PRINTER."
150 PRINT:P1$="L255O5ABAFAGFDEDACABAFAEDAO4ABAFADCCO1ABADDEO5ABADD"
160 PRINT" - PRESS THE [SPACEBAR] NOW  -"
170 I$=INKEY$
180 IF I$<>CHR$(32) THEN GOTO 170 ELSE PLAY P1$
190 FOR HOLD=i TO 1000:NEXT HOLD
200 CLS
210 PRINT:PRINT" [1] = YOU REQUIRE INSTRUCTIONS."
220 PRINT:PRINT" [2] = YOUR HI-RES SCREEN HAS              ALREADY BEEN LOADED IN."
230 PRINT:PRINT" [3] = YOU DO NOT REQUIRE THE              INSTUCTIONS."
240 PRINT
250 INPUT" WHICH [1/2/3] - ";IN$
260 IF IN$<>"1" AND IN$<>"2" AND IN$<>"3" THEN GOTO 200
270 IF IN$="1" THEN FOR HOLD=0 TO 1000:NEXT HOLD:GOTO 310
280 IF IN$="2" OR IN$="3"THEN FOR HOLD=0 TO 1000:NEXT HOLD:GOTO 510
290 REM INSTRUCTIONS FOR
300 REM SCREEN DUMP
310 CLS
320 PRINTTAB(4)"----------------------"
330 PRINTTAB(4)"seikosha screen dumper":PRINTTAB(4)"----------------------"
340 PRINTTAB(7)" -instructions-"
350 PRINTTAB(7)" ---------------"
360 PRINT:PRINT" TO USE S.S.D. ALL YOU HAVE TO    DO IS TO LOAD IN YOUR HI-RES
 SCREEN FROM TAPE WHEN ASKED ,   ";
370 PRINT" SET UP (WITH PAPER READY) AND    SWITCH ON YOUR PRINTER AND AND  LET T
HIS PROGRAM DO THE REST!"
380 PRINT:PRINT" - PRESS THE [SPACEBAR] NOW -"
390 I$=INKEY$
400 IF I$<>CHR$(32) THEN GOTO 390
410 PLAY P1$
420 FOR HOLD=0 TO 1000:NEXT HOLD:CLS
430 PRINT" **************"
440 PRINT" note-important"
450 PRINT" **************"
460 PRINT:PRINT" IF THIS PROGRAM IS newed WHEN    THE HI-RES SCREEN HAS LOADED ,
 FOLLOW THE INSTRUCTIONS (a/b)    IN THE INSTRUCTIONS."
470 PRINT:PRINT" - PRESS THE [SPACEBAR] NOW -"
480 I$=INKEY$
490 IF I$<>CHR$(32) THEN GOTO 480
500 PLAY P1$:FOR HOLD=0 TO 1000:NEXT HOLD
510 REM ** MAIN PROGRAM **
```

Dump C — cont.

```
520 CLS
530 PRINT"  ----------------------------"
540 PRINT"  * seikosha screen dumper *"
550 PRINT"  ----------------------------"
560 PRINT:PRINT" WHICH pmode (0/2/4) WAS YOUR    HI-RES SCREEN SAVED IN-";:INPUT
P
570 IF P<>0 AND P<>2 AND P<>4 THEN PLAY P1$:GOTO 510
580 PRINT:INPUT" DO YOU WANT A small OR large    SCREEN DUMP (S/L)";SZ$
590 IF SZ$<>"S" AND SZ$<>"L" THEN PLAY P1$:GOTO 510
600 IF IN$="2" THEN GOTO 720
610 PRINT:PRINT"  - POSTITION YOUR TAPE NOW -"
620 MOTOR ON:AUDIOON:PRINT:PRINT" - PRESS THE [SPACEBAR] WHEN -   - YOUR TAPE IS
 POSTITIONED! -"
630 SOUND 100,2
640 I$=INKEY$:IF I$<>CHR$(32) THEN 640
650 MOTOROFF:AUDIOOFF:SOUND 100,2
660 CLS:PRINT:PRINT" PRESS play ON YOUR CASSETTE         PLAYER AND PRESS ANY KEY...
..."
670 I$=INKEY$
680 IF IN$<>"" THEN 690 ELSE GOTO 670
690 PMODE P:SCREEN1,0:CLOADM:SCREEN1,0:
700 SOUND 100,5
710 PRINT:PRINT" ...loaded...loaded...loaded..."
720 PLAY P1$
730 PRINT:PRINT" PRESS ANY KEY WHEN YOUR PRINTER        IS READY TO PRIN"
740 I$=INKEY$:IF I$<>"" THEN 750 ELSE 740
750 IF SZ$="S" THEN GOTO 780
760 IF SZ$="L" THEN GOTO 960
770 PRINT:PRINT"error - A ERROR HAS BEEN MADE    SOMEWHERE WHEN INPUTING AND IT
HAS GONE PAST THE VALIDATION     CHECK...THE PROGRAM WILL NOW BE re-run...SORRY."
780 ' SMALL SCREEN DUMP ROUTINE
790 PRINT£-2 , CHR$(13)
800 PMODEP,1:SCREEN1,0
810 PRINT £-2 , CHR$ (8)
820 FOR X=0 TO 191 STEP 7
830 FOR Y=0 TO 255
840 A=PPOINT(Y,X)+PPOINT(Y,X+1)*2+PPOINT(Y,X+2)*4+PPOINT(Y,X+3)*8+PPOINT(Y,X+4)*
16+PPOINT(Y,X+5)*32+PPOINT(Y,X+6)*64+128
850 PRINT£-2,CHR$(A);
860 NEXT Y
870 PRINT£-2,CHR$(13);
880 PRINT£-2,CHR$(8);
890 NEXT X
900 PRINT£-2,CHR$(15)
910 CLS:PRINT:PRINT" f i n i s h e d ! ...o.k..."
920 PRINT:INPUT" DO YOU WANT ANOTHER COPY (Y/N)";AC$:IF AC$<>"Y" AND AC$<>"N" TH
EN SOUND 100,2:GOTO 910
930 IF AC$="N" THEN PLAY P1$:RUN
940 IF AC$="Y" THEN PRINT:PRINT"press any key to start printing!";
950 I$=INKEY$:IF I$<>"" THEN 780 ELSE 950
960 ' LARGE SCREEN DUMP ROUTINE
970 PRINT£-2 , CHR$(13)
980 PMODEP,1:SCREEN1,0
990 PRINT£-2,CHR$(8)
1000 FOR X=255 TO 0 STEP -7
1010 FOR Y=0 TO 191
1020 A=PPOINT(X,Y)+PPOINT(X,Y)*2+PPOINT(X-1,Y)*4+PPOINT(X-1,Y)*8+PPOINT(X-2,Y)*1
6+PPOINT(X-2,Y)*32+PPOINT(X-3,Y)*64+128
1030 PRINT£-2,CHR$(A);CHR$(A);
1040 NEXT Y
1050 PRINT£-2,CHR$(13);
1060 PRINT£-2,CHR$(8);
1070 FOR Y=0 TO 191
1080 IF X>3 THEN A=PPOINT(X-3,Y)+PPOINT(X-4,Y)*2+PPOINT(X-4,Y)*4+PPOINT(X-5,Y)*8
+PPOINT(X-5,Y)*16+PPOINT(X-6,Y)*32+PPOINT(X-6,Y)*64+128 ELSE A=PPOINT(X-3,Y)+128
1090 PRINT£-2,CHR$(A);CHR$(A);
1100 NEXT Y
1110 PRINT£-2,CHR$(13);
1120 NEXT X
1130 PRINT£-2,CHR$(15)
1140 CLS:PRINT:PRINT" f i n i s h e d ! ...o.k..."
1150 PRINT:INPUT" DO YOU WANT ANOTHER COPY (Y/N)";AC$
1160 IF AC$<>"Y" AND AC$<>"N" THEN SOUND 100,5:GOTO 1140
1170 IF AC$="N" THEN PLAY P1$:RUN
1180 IF AC$="Y" THEN PRINT"press to start printing....":
1190 I$=INKEY$:IF I$<>"" THEN 960 ELSE 1190
```

Dragon Comms

Martyn Armitage introduces communication for the Dragon

SO, you've got a Dragon 64 and the only socket that you've probably never used is the one marked up Serial 1/0, the RS232 interface to give it it's real title. There are two main uses for the interface, 1) for connecting a serial printer, and 2) for connecting a modem. Most of what is said in this article will be regarding the use of the port as a communications interface, along with a modem, but most of what is written will also hold true for the connections of a printer. To start off with I shall attempt to put into layman's English some of the most commonly used jargon that is associated with the RS232 interface, such as parity, baud etc. First we'll start off with protocol, which itself can, and will , be split into sections.

Setting standards

Protocol can be best be described as communications 'etiquette', and is simply a set of standards that one should adhere to if you wish to get anything other than garbage on the screen (paper if you're using a printer). As I have already said, I am going to break this heading down into the various parts that it consists of.

PARITY is a very simple method of error detection on received data. When in the communications world you will hear the phrases 'odd parity' and even 'even parity' quite regularly, and just as often you will hear 'no parity'. Just what is parity, and how does it work? As you will know, a byte is made up of eight bits, each bit being either set (1) or clear (0). Parity works by counting the number of set bits (1s) in a byte of data, and either, setting or clearing the 'parity bit' in the transmitted data. Odd parity used the parity bit to make the number of set bits 'odd'. For example, if we were to transmit the character A, it's bit pattern is 01000001, and as you can see ther are an even number of set bits, (2). So with odd parity the parity bit would be set to make an odd number, (3). If there is already an odd number of set bits then the parity bit is left clear.

The same thing still happens under even parity, except that the parity bit is used to make an 'even' number of set bits. The error detection works in this way: when data is received, the set bits are counted. If it is found that there is an even number of set bits, and odd parity is in use, then obviously there is an error somewhere in the byte. There is no indication as to which bit or bits have been affected, only that there is an error. It can be seen from this, that parity checking used on its own is of very little use, and in fact the use of no parity, where no parity checking is done, is quite widely used.

Mark/space

MARK and SPACE: when we talk of bits within a byte we usually say that they are either set or clear. When talking in terms of communications a set bit is termed a mark, and a clear bit is given an term space.

START/STOP BITS: there are two types of transmission of data, synchronous and asynchronous. As the interface for the Dragon 64 is the 6551 Aysnchronous Communication Interface Adaptor, ACIA for short, and is not capable of true synchronous transmission, I shall simply say that with synchronous methods, the two communicating computers are synchronised with each other and so once synchronised with each other there is no need to indicate where each byte of data starts or stops. A series of synchronisation bytes being transmitted at intervals, very much similar to the methods used for saving programs etc on tape. The 6551, being asynchronous, has to use other methods of determining the start and end of a data byte, this is done by the use of start and stop bits. A start is simply a bit that indicates the start of data. The start bit is transmitted as a mark, then come the data bytes, which are obviously mark or space depending on the bit being transmitted. They are then followed by the stop bit(s), which like the start bit is transmitted by the mark. As you can see an eight bit byte, when transmitted by the 6551 asynchronous interface adaptor, is at least ten bits long. Depending upon the protocol being used either one or two stop bits are used

BAUD RATE: the baud rate is a measure of speed of tranmission, and can be translated to 'bits per second'. For example 300 baud can be regarded as 300 bits per second, or 37.5 bytes per second (300/8 = 37.5) This figure of 37.5 bytes is of course assuming the transmission of 8 bits per byte, but as I have said above we are transmitting at the least 10 bits per byte, and so 300 baud becomes 30 bytes/characters per second (300/10 = 30).

SPLIT BAUD RATE: a split baud rate is a way of saying transmitting and recieving data at different speeds. The standard split baud rate is 1200/75 and 75/1200. The first number is the speed that the receiver is working at and the second is the speed that the transmitter is working at. In the first case the data is received at 1200 baud and transmitted at 75 baud, the second case is the opposite, receive at 75 baud and transmit at 1200 baud.

FULL/HALF DUPLEX: full duplex describes the abillity of the ACIA to communicate in both directions at the same time, whereas Half duplex describes the ability to transmit and receive but at the same time. When using the half duplex method it is necessary for each machine to let the other know when it has finished sending and is ready to receive. This is very much like two people communicating over a two way radio, and having to say 'OVER' etc. Full duplex is like a telephone conversation, each can talk at the same time.

Internal checks

That just about covers most of the most commonly used phrases and terms. The user of the 6551 should realise that all the parity checking, addition and subtraction of the start and stop bits is carried out internally by the chip and any errors are indicated by the flags in the status register of the chip. The data is presented to the user in the same condition as it was transmitted, unless of course an error has occurred, in which case the data could be just about anything.

The RS232 interface does not operate on TTL levels (0V to 5V) for signalling but instead uses voltage levels between -25 volt and +25 volt. The Drgons interface uses what appears to be the norm, namely -12V to +12V. One other thing with the RS232 interface is that on the data lines a voltage level of between +5V and +25V represents a lo (0) and -5V to -25V represents a hi (1), with the opposite being true for the control lines.

The computer is given the term Data Terminal Equipment (DTE) and the modem is given the name Data Communication Equipment, or DCE for short.

Now onto the Dragons RS232 port. You probably won't be surprised when I say this but, the Dragons serial port looks as though it was bodged onto the machine, the user being provided with two of the five possible control lines available, however, the control lines provided do allow the interface to be used in a satisfactory manner. There are seven connections available, they are: 1) TX, 2) GND, 3) DTR, 4) RX, 5) CTS, 6) +12V, 7) -12V. Lets go into a little bit of detail.

TX is the abbreviation for transmit, all the data being sent by the Dragon leaves on this pin.

GND is the ground pin and is udes as a return for all of the signal lines. RX, is the receive data pin, here all the data comes into the Dragon. DTR, which is short for Data Terminal Ready, is the line used by the computer (DTE) to indicate to the modem

(DCE) that it is Ready and in a state to receive data. Next we come to the CTS (Clear To Send) pin. This line is used by the modem to indicate to the computer that it is in a condition to receive data, and then send it outwards to the telephone line. It is essential that this line is connected on the Dragon otherwise you will find transmission impossible although you will be able to receive data.

Connection

The last two pins, +12V and -12V, are provided in order to drive the control lines hi or lo, and are not capable of providing power sources for any other use.

CONNECTING TO A MODEM: of the five lines that the serial port provides the least number that we can get away with connecting is three. They are TX, RX, and GND. If you're alert then you should be saying 'But what about CTS? It's impossible to transmit without that connected.' Well noticed. Yes we do have to connect to CTS, but we can fool the Dragon. There are two methods open to us; 1, we can connect the -12V pin to CTS, which will then inform the 6551 that it is always clear to send, or we can connect DTR to CTS, which will inform the 6551 that is clear to send when ever the DTR line is low, ie whenever the Dragon is on line. The links between the pins should be made inside the 7 pin plug that is on the end of your modem cable, and not inside the Dragon itself. When connecting to the modem it should be realised that the TX and RX connections should be reversed, that is the TX of the Dragon should be connected to the RX of the modem, and the TX of the modem should be connected to the RX of the Dragon. The GND pins should be connected to each other. The above is the minimum connections that we can make. If your modem has extra control lines then it makes sense to make use of them. I shall illustrate all of the connections possible from the Dragon's port.

Dragon	Modem	Modem	Modem
GND	GND		
TX	RX		
RX	TX		
DTR	DTR	RTS	
CTS	CTS	DSR	DCD

The connections in the first two columns are the recommended ones but some modems, like the Dragon, do not possess all of the connections possible under the RS232 interface standards, and so the connections in the third and fourth columns are alternatives if the previous option isn't available.

GETTING ON LINE is the final stage. We've connected the Dragon to the modem, the modems connected to the phone line, what next? Well before doing anything we must make sure that we are set up using the same protocol as the computer that we are going to call. Most Bulletin Boards (BBS) use 300/300 baud (transmit and recive at 300 baud), 1 start bit, seven data bits, even parity, 1 stopbit, using this protocol should give you some measure of success in most cases. On the increase is the use of 1200/75 baud rate, this is receiving data at 1200 baud and transmitting data at 75 baud. The 6551 as used on the Dragon 64 is not capable of working full duplex at split baud rates, but it is posssible all the same, to access computers running at split baud rates, for example Prestel.

Conclusion

The above is a very basic introduction to what is a very interesting subject, many books have been written about communications, although none of them, to my knowledge, specific to the Dragon. If there is enough demand and the editor is in agreement, then I shall attempt to write further articles which will go into more detail, and possibly dedicate one to programming the 6551 (in machine code) so that you could write your own personal communications package.

Adventure Contact

To help puzzled adventurers further, we are instituting an Adventure Helpline — simply fill in the coupon below, stating the name of the adventure, your problem and your name and address, and send it to Dragon User Adventure Helpine, 12/13 Little Newport Street, London WC2H 7PP. As soon as enough entries have arrived, we will start printing them in the magazine.

Don't worry — you'll still have Adventure Trial to write to as well!

Adventure ....

Problem ....

....

Name ....

Address ....

....

IF any reader does have serious delivery/non delivery problems with any supplier, whether or not they advertise in DU, we would like to know. Only rarely can we do anything to improve a genuinely dodgy situation (which fortunately are rare) but it helps us to build up a profile and identify any long-term problems. Come to think of it, that includes *Dragon User*, as, naturally, we want to know about any bottlenecks as early as possible. Apologies in advance for not acknowledging every letter, but where we *can* be of practical help, we will.

Winners and Losers

Every month Gordon Lee will look at some prize programming

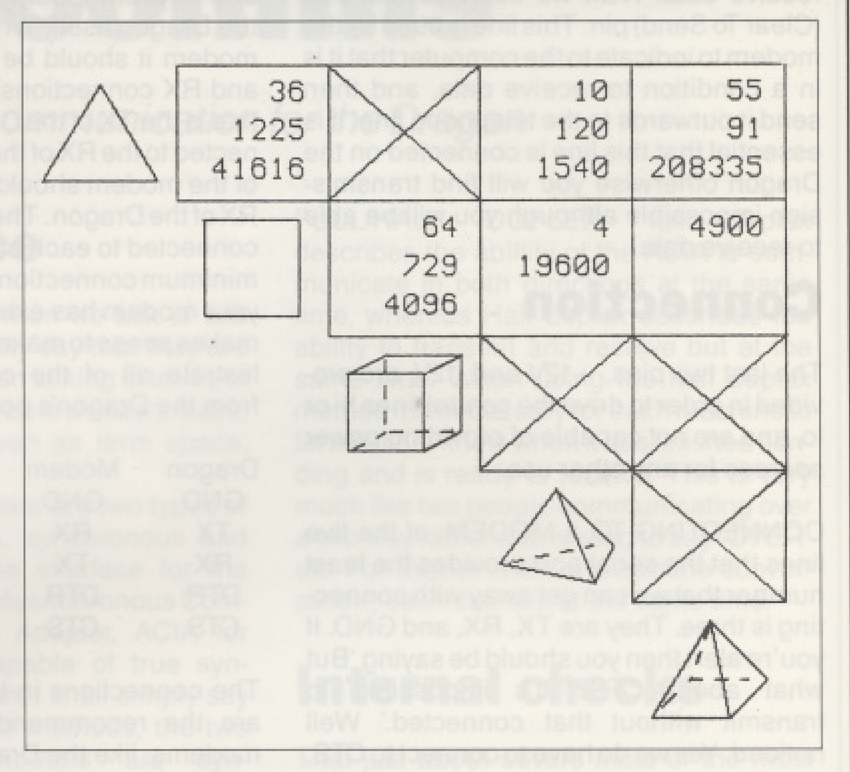
IN January's *Dragon User* I game my solution to the October compeition problem. This was to find certain integers which belong to more than one class of the figurate numbers. To keep the solution as simple as possible, the listing that was given compares just two of the types of figurates at a time. In the example given, the comparison was between the square and triangular numbers. When this program is run, each matching value is printed in turn and the program can be stopped as soon as the required number of values has been found. The formulae used in lines 20 and 30 of last month's program are those which were in the table in October's issue. By amending one or both of these formulae between each run, all of the six required pairings can be computed. The actual method that the program uses, which utilises a 'leapfrog' technique to prevent unnecessary duplication of calculation, was outline last month. This speeds up the execution time of the program.

Listings

I was therefore surprised at the length of some of the entries received — quite a number having a listing with over a hundred program lines — too long to be reproduced here. In considering some of the shorter listings, was surprised to find that some, when tested, failed to produce the solutions submitted with them!

Function

However, one which did come up with the right answer is given here. It was from E A Newman of Addlestone, Surrey, and is interesting in that it makes use of the function (FN) command of the Dragon. The five formulae which produce the triangular, square, cubic, tetrahedral and pyramidal numbers are initially defined in lines 10 and 20. To perform any of these calculations subsequently in the program it is simply necessary to use a line such as X=FNTR(N), which, if N is at that moment, say, 5, will compute X as 15 — the fifth triangular number. The advantage of using the FN command is that it can be used repeatedly in different parts of the program without the need to type in the generating formulae more than once. In the case of particularly complex formulae this can prove quite an advantage, and will also help to minimise errors! In E A Newman's listing the pairs of figurates required at each stage of the computation are calculated, using the FN command, in the subroutine at lines 30 to 80. In lines 90 and 150 is a motley selection of IFs, THENs and ELSEs. Normally I would not recommend the use of more than one set of these commands in a single program line unless you fully understand the logic which the computer uses in interpreting these lines (in the same way that a too liberal use of ANDs and ORs can also produce confusing results!).

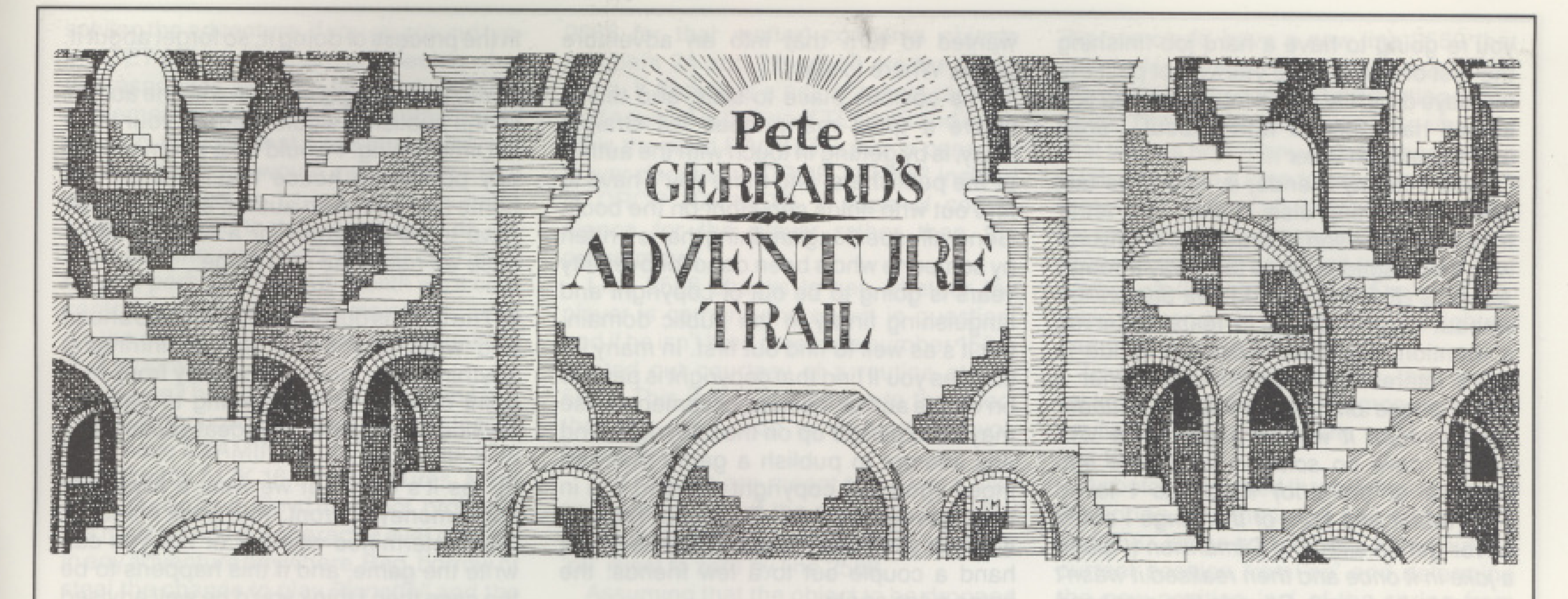


Austin Henderson's solution, using a Tandy CGP-115 printer.

A number of other readers also included a printer routine in their programs to reproduce the grid arrangement for the answers. Thanks to Nils Lindgren of Waxholm, Sweden, and Randy Longshore of Chesterfield, USA, who were among the winners who presented their solutions graphically. Special mention must also be made of Austan Henderson of Bromsgrove for a very neat piece of graphics produced using an inexpensive Tandy CGP-115 printer.

```
10 CLS:DIM R(5,3):DEF FNTR(X)=X*(X+1)/2:DEF FNSQ(X)=X*X:DEF FNCU(X)=
   X*X*X:PL=-1
20 DEF FNTH(X)=X*(X+1)*(X+2)/6:DEF FNPY(X)=X*(X+1)*(2*X+1)/6:GOTO 90
30 X=FNTR(N):Y=FNSQ(M):RETURN
40 X=FNTR(N):Y=FNTH(M):RETURN
50 X=FNTR(N):Y=FNPY(M):RETURN
60 X=FNSQ(N):Y=FNCU(M):RETURN
70 X=FNSQ(N):Y=FNTH(M):RETURN
80 X=FNSQ(N):Y=FNPY(M):RETURN
90 T=2:N=2:M=2:PL=PL+1:IF PL=6 THEN GOTO150 ELSE IF PL=4 THEN F=1 EL
   SE IF PL=5 THEN F=2 ELSE F=0
100 ON PL+1 GOSUB 30,40,50,60,70,80
110 IF X=Y THEN R(PL,T)=X:GOTO 140
120 IF X<Y THEN N=N+1 ELSE M=M+1
130 GOTO 100
140 IF T=F THEN GOTO 90 ELSE T=T-1:N=N+1:GOTO 100
150 FOR I=0 TO 5:IF I<1 THEN K=0:U=I ELSE IF I<3 THEN K=0:U=I+1 ELSE
    K=4:U=I-2
160 FOR J=0 TO 2:V=2-J:IF R(I,V)<>0 THEN PRINT@32*(J+K)+7*U,R(I,V)
170 NEXT J,I:PRINT@400
```

EA Newman's program uses the function command.



SLIGHT panic at chez Gerrard as he loses his master file complete with numerous Dragon adventure solutions, but panic over as it is retrieved from beneath two copies of *Dragon User*, two programming utilities with manuals the size of telephone directories, and a Spanish edition of one of my books. I must sort out this filing system of mine some day.

A belated thank you to the people who sent me Christmas cards via *Dragon User* — oh, I didn't know you cared! A special thank you to Joe Brincat, our noted correspondent from Malta, who sent me not only a Christmas card but a postal order for one pound so I could buy myself a drink. What a star. I must point out, by the way, that when I gave Joe's address last time I got things slightly wrong. The full address is 73 Annunciation Street, Hamrun, Malta. What I printed as the post code last time was, in fact, the date! Foolish boy, slapped wrist, apparently all the people who wrote to Joe (I pass on his thanks) included that as the post code, much to the amusement of Joe and his postman.

It seems that many of you wrote to him, exchanging cassettes and giving details of many games and utilities. Some, it seems, had well over 200 titles on offer. Well done chaps and chapesses, keep up the good work.

Keeping briefly on the foreign side of things, Ola Eldoy (who can be found at Stokken, N-5410 Sagvag, Norway — postcode not being confused with date this time) asks me to tell you that he runs a Norwegian Dragon magazine, and that he can get help for the following adventures: *Black Sanctum, Calixto Island, Castle of Doom, Caverns of Doom, Circus Adventure, Don't Panic, Dragon Mountain, Franklin's Tomb, Golden Baton* (easy-peasy, solved it myself in one afternoon!), *Juxtaposition, Mansion Adventure, Mansion of Doom* (fond of Doom, these adventure writers), *Pettigrew's Diary, Pirate Adventure, Return of the Ring, Ring of Darkness (and rings), Sea Quest, Shenanigans, The Ket Trilogy, Time Machine, Trekboer*, and finally the *Vortex Factor.* If you fancy a few exotic stamps, drop him a line.

Picks letter from pile, finds out that it's from Sean Neale in Wakefield, who says that I write a superb column. Who is this man, get his query answered immediately. It's an old friend, is *Syzygy*, and I've got yet another letter here (from one Steven Wood in Essex — we shall be returning to this chap later) about the same problem in the same game. It's all to do with entering those co-ordinates to whizz you about all over the place once you've got to the transporter. Well, as we all should know by now, the co-ordinates for the various places that you want to get to are:

Planet 0-4-1-5
Emerald 2-7-3-0
Vader 1-6-0-3

Read this carefully, Sean and Steven: for each place, enter each number by pressing that number, and at the end of the row (after the fourth number has been entered in other words) you must PULL LEVER. And away you go. I thought nobody would ever have problems with *Syzygy* ever again, but it just goes to prove that this extremely popular adventure for the Dragon still manages to confuse people. The sign of a good game, I suppose.

Problems with *Tanglewood* for Andrew Lamb, in Hazel Grove, Stockport, Cheshire. An address that should be familiar to a few of you ... but I shall say no more, she who must be obeyed would probably ban this column if I mentioned another publishing company. Any road up, as they say, he's having problems getting the archangel, despite (and I quote) many, many hours of trying. Well, section 5 of the mega-hint sheet, the part that deals with the collecting of the archangel, reads as follows:

"Send Bruce into Dwarf Dive and press the button to reset the lifts. Bring him out via the northern exit/entrance. Send Foghorn in through the northern entrance and down on the lifts to get the whiskers. Bring him out via either the northern or southern exit and transfer the whiskers to Goliath. Get Goliath to wear the whiskers. To complete Dwarf Dive, send Goliath in, down on the lifts and ladders and get the archangel from the lowest mushroom and give the archangel to Beanbag. This is hard to complete without getting caught, so use the HOLD and QUIT commands."

There, let's hope that's sorted that particular problem. A little bit further on in the game, Andrew tells us that he can't get past the walled garden. I shall quote step number 11 in the solution:

"Send Goliath into the walled garden carrying the wand and Cast Spell then get the rod. He returns from the ride with the wand at daybreak. Use the wand to fish for the specs and give them to Foghorn to wear."

That's all we need to know about *Tanglewood* I think, so back to Stephen Wood and his many problems. All of them, I hasten to add, are connected with adventure games. Wouldn't want anyone to get the wrong idea ...

Where should we start, Stephen "llaw eht tsniaga daeh ym gnignab llits tub emit tsal sa kcuts sa etiuq ton" Wood? And for printing that, I win the bet. Anyway, let's take a look at *Trekboer.* Again. Where's the Xendos flower, he wants to know. On the garden planet, as last month's column will tell you. Where's the acid? North from the ship on 8350. Where's (love this!) the thing to help you breathe on planet? So technical, a suit is the object in question I believe. Have we unlocked the cabinet? Have we opened the manual? Have we pressed the red button in the teleport room? Are we, in fact, wasting our time attempting to play adventures and should we do something marginally easier instead, like unfolding paper clips or sharpening pencils? Hee hee, sorry Stephen, couldn't resist a wee dig at someone. Where are other useful objects, he would like to be informed. Well, I can't tell you everything, now can I? Read last month's column in minute detail, it's all in there somewhere.

And onto *Juxtaposition*, where the question is akin to something out of *The Restaurant at the End of the Universe*: where's the spade with which to dig in the loose earth past the door past the nighteye droid so as to get the Red Cetite Ore? Your hovercraft is full of eels, and all that. Ayup lad, what tha must do is to blind the nighteye droid with your camera and take the spade from him. Where's the camera? In the photo gallery. Where's the photo gallery? Go to the monorail platform (don't enter endless rez zones!), wait for the car and board it when the doors open. And if you can't find the monorail platform then

you're going to have a hard job finishing the rest of it! No, once you've got past the nighteye droid life gets a lot easier, and you should have little to worry about. I shall await your next letter ...

Now this, my friends, is where we talk about something else, thanks to a letter from Paul Harrison in Shropshire. Anyone who's managed to wade their way through Tolkein's (*Tolkien's — Ed.*) *The Silmarillion* without getting bored to tears deserves a mention. Our local postcode guide is more interesting. (*Rumour has it that a game based on The Silmarillion is under development. It will be played on a VAX cluster, have no solution, and slide into the sea at the end*) Much as I loved *The Hobbit* and *Lord of the Rings* I could never get on with *The Silmarillion* (*I found a joke in it once and then realised it wasn't meant to be funny*). Sounds like a spaced out heavy metal band to me, but there you go. (*It is. I was up till 2am last night watching them, which may explain my present state of mind. What d'you call a 6 foot 4 inch Scotsman wearing make up and a mini dress? 'Sir', of course.*)

Paul's first comment is a novel one: have a competition where someone sets up a title and anyone with nothing better to do tries to make an adventure out of it. Okay, this month's title is: Gerbil Riot of '47.

Secondly, he goes on to say that he would like to write an adventure based on *The Silmarillion* and other such books and would like to know how to start going about it. Well, first of all there's a teeny little thing called The Law of Copyright...(*There's a terrific title!*)

To take a specific but non-real (I hope!) example, suppose that you'd just finished reading *The Price of Eggs* by Dimli Gloing, published by TrollSlayer Press, and wanted to turn that into an adventure game. Where to start?

The obvious place to start, and this is where a great many ideas fail straight away, is by getting in touch with the author or the publishers, and there you'll have to find out who holds copyright on the book. Generally speaking anything that is written by someone who's been dead for over fifty years is going to be out of copyright and languishing firmly in the public domain, but it's as well to find out first. In many instances you'll find that copyright is passed on via the author's estate, or someone else may have picked up on the copyright, and any attempt to publish a game that infringes upon that copyright will land you in very severe trouble. This can happen whether you sell a thousand copies or just hand a couple out to a few friends: the same rules apply.

To begin with, then, you must write to the publishers. Don't bother with the author at this stage. For one thing they might be dead, and for another they might have sold all copyright to the publishers when it was released. So in our case a letter to TrollSlayer Press is the place to start. Several things can then happen.

1) They have only published the British version of an American book, in which case you're in for a long correspondence with the original publishers, and this can also happen if it's a paperback edition of a book originally published in hardback, as quite often two different companies take charge of the same book. When you eventually reach the first publisher, we can go on to the next stage.

2) Somebody else has already done it, or is in the process of doing it, so forget about it.

3) You'll have to get in touch with the author, so the publisher would pass on your letter to Dimli Gloing. I would take a dim view of any publishing house that released the home address of an author, and you'll just have to be prepared for a wait. In either case we reach the next stage.

4) The thing is out of copyright, so go ahead and write your game without infringing anything and, more importantly from your point of view, without having to pay any royalties to anyone. The ideal situation.

5) Yes it's okay but we want a royalty or some money up front. Your bank manager then determines whether or not you can write the game, and if this happens to be the case then I think you'd be well advised to give up on the idea and find something else to write about.

6) No it is not okay, it is completely out of the question, and don't even think about it. This quite possibly means that the publishers are doing it themselves, or having received your idea think it a wonderful one but want a massive software house to do it. Oh well, nice try, but back to the library.

If you actually survive all these traumas and receive permission to write the game (whether you're paying a royalty or not, and I'd strongly advise, knowing that I'm repeating myself, not to take on a game that requires you to pay money to someone else: you're better off out of it), then what do you do?

Since I've run out of space, you'll just have to buy next month's *Dragon User*!

Write: ADVENTURE

***Pete Gerrard** discusses literary decorum and the joys of spring*

ONE word which has entered the adventure player's vocabulary lately, and which certainly wasn't seen in any of the early adventures, is the word 'all'. People now frequently try and GET ALL or DROP ALL, whereas before they were content to get or drop things individually. This is part of the increasing power of the parser and the demand for more sophisticated games.

Anyone who has played, for example, *The Pawn*, on another computer, will know what I mean. From my own point of view I cannot see the point in having to type in the command Put the pot plant in the plant pot, however wonderful it may make the parser appear. In this case I'm sure it was only ever included so as to impress gullible reviewers, for the blessed pot plant is in the plant pot to begin with! You have to take it out, using a trowel if I remember rightly, and then put it back in again, all this to score about 5 points. A complete and utter waste of time, but Magnetic Scrolls like to boast about their parser, so there you go.

Again from my own point of view, however sophisticated the parser might be I still tend to play an adventure in verb ... noun format, only occasionally using more words where a situation clearly requires it. Why on earth type Climb the slippery rope wearing the stout gloves when you could just as easily have Wear Gloves, Climb Rope instead? Less typing, and less time for the program to have to work out what you mean. Perhaps I'm in a minority, but personally I think that a parser that understands four words is sufficient, and I absolutely loathe and detest adventures that insist that you (for example) go north quietly, or untie rope carefully. If it was a real-life situation rather than an adventure, if you had to go north quietly you WOULD go north quietly, you wouldn't trample in like a wounded elephant, and I think that programmers should accept that those things are implicit in the instruction given to the program. An adventure should be about solving problems, not working your way through *Roget's Thesaurus* in a vain attempt to find the correct word.

What's brought all this on, you might be wondering. Hangover? No. Sound thrashing in last night's pool league match? No. It was a quote written by a friend of mine, which read that a definition of frustration was spending three months typing "Say 'give life with thee my trusty blade"' near the end of an adventure only to find (quite by accident) that the correct input is "Shout 'give life with thee my trusty blade"'. That sort of thing should never happen, so keep it out of your own adventures, eh?

However, I can appreciate the usefulness of the word ALL. Not only does it save time, it can also be a useful aid to

solving the adventure, if you go around typing GET ALL everywhere and seeing what the responses are. The first time I tried to implement such a routine in an adventure I gave up, thinking that it was impossibly difficult and nobody would ever want to use it anyway. However, I tried again recently, and discovered that it wasn't quite as difficult as I thought. We'll look at GET next month, perhaps, but this month we'll concentrate on DROP. This is mainly because it's the easier one of the two to understand and program!

We won't be going any further than GET or DROP though. Even the mighty Infocom don't have EXAMINE ALL, for instance, in their mighty 512K 16-bit adventures. Mind you, they do have Floyd the robot, probably the most wonderful character ever created in any adventure anywhere. Beg, borrow or steal the chance to play *Planetfall* and the even better follow-up *Stationfall*. The creation of Floyd was a work of genius, and whoever thought of him first deserves an adventurers' medal. Gales of laughter one minute, then silent tears at the end of Stationfall. A truly memorable character, and quite, quite real and believable.

If you look at the listing headed **Figure one** you will see a fairly conventional DROP routine from one of my adventures, and in order to understand it I'd better explain what a few of the variables are.

Lines 2550 and 2551 concern themselves with object number 61, a dog. This was a different slant on a conventional idea, because the dog is in fact a guide dog, and without him you cannot get through the blackness of the cave section of the game. It makes a change from roaming around trying to find a lamp and some batteries, a situation I am somewhat tired of seeing. If you're dropping the dog then tell the player so, and if he's not carrying object number 11 in the correct manner (a candle in a box, to stop the wind blowing it out) then the variable 'lo' is set to zero, indicating light off.

Line 2552 sorts out link words and nouns by going to line 2556 if a link word has been typed in. The player might have typed DROP BOX ON FLOOR, which we just interpret as DROP BOX after lines 2566 and 2568 have had their say. Line 2554 (and 2566 for that matter) concerns objects numbers 24 and 26, which are the same thing. It (they?) is a pair of tennis rackets, and noun 24 is the word TENNIS and noun 26 is the word RACKETS, just in case the player decides to DROP TENNIS instead of DROP RACKETS. A case of convenience for the player rather than the programmer.

Line 2556 checks to see whether the player is carrying the object in question, and if he isn't then message number 153 is printed out courtesy of a routine at line 5995. This is just something to the effect that you can't drop something that you haven't got.

The rackets are meant to be worn, like snowshoes, and if the player is dropping the rackets then the 'rackets worn' variable 'rw' is set to zero by line 2558.

Assuming that the object to be dropped isn't object number 42 (the number is mere coincidence, I assure you, and nothing to do with the meaning of life!) then line 2560 sorts everything out by placing the object in the current location and informing the player that he has, in fact, dropped it.

Lines 2562, 2564 and 2570 are all designed with object number 42 in mind, and object 42 is in fact a tightly coiled spring. Reasonably enough, dropping such an object causes it to bounce away from you, so we use the current position variable 'cp' to determine the new position ('np') of the spring after it's finished its bouncing. Then we put the spring in its new position and, as with all the 'drop' routines, decrement the counter that keeps track of how many objects the player is carrying. This, for some obscure historical reason that now escapes me, is the variable.

That was the conventional 'drop' routine, so how might we go about turning it around so that we can understand DROP ALL as well as DROP OBJECT? Well, if you look at the listing in **figure two** you'll see just one way of doing it. If you think about it, we're only really concerned with three objects, namely objects numbers 26, 42, and 61. The rest can be dropped with wild abandon, but dropping those three causes something to happen, so as long as we keep that in mind we should be all right.

Assuming a hasty bit of line renumbering, we could have a new line 2550 that checks for the word 'all'. If it's present, then trundle off to our new routine starting at line 2580. There, with 'nn' being the variable that tells us how many objects there are in total, we set up a loop to scan for every object in turn. First of all, in line 2582, we look for the dog being present, and if it is then proceed as before and tell the player that it's been left behind. Again, if he's not carrying the candle in the box then the light on variable 'lo' must be set to zero.

In line 2584 we look for the tennis rackets, and if they're being carried then we have to set the rackets worn variable 'rw' to zero.

As before, we need three lines to cater for the spring, and these are now 2585, 2586 and 2587. Same sort of thing, found out the current position from 'cp' and determine the new position 'np' of the spring from that, and print out the 'Boinnnggg!!!' message to inform the player that something's happened to the spring. Incidentally, one of the more annoying features when testing the program that these listings come from was the spring, for I had included a routine that made it automatically hop away from you every ten moves, regardless of whether you had dropped it or not. I spent ages chasing the thing all over the game, and just when I thought I'd caught up with it after solving a problem or two it would hop away again. It might only be moving two locations at a time, but those two locations are not necessarily anywhere near each other!

Line 2587 checks to see whether an 'ordinary' object is in the player's possession, and if it is then cast it to the floor it tells player what's been dropped before continuing around the loop with the NEXT statement in line 2588. Line 2589 then sends us back to the main part of the program, and you might care to expand on this section of the program slightly so that it would cater for an input of DROP ALL when the player wasn't actually carrying anything. Perhaps I think too highly of players, and neither of my testers found it, but someone, somewhere, would no doubt type it in!

So, DROP ALL wasn't too difficult, and we'll take a look at a GET ALL routine next month. Bye for now.

Figure one

```
2550 IF na=61 AND ob(61)=-1 THEN PRINT "
You leave the dog.":ob(61)=cp:zz=zz-1:IF
 ob(11)<>-2 THEN lo=0:GOTO 10
2551 IF na=61 AND ob(61)=cp THEN 10
2552 IF li<>0 THEN 2566
2554 IF na=24 THEN na=26
2556 IF ob(na)<>-1 THEN mess=153:GOTO 59
95
2558 IF na=26 THEN rw=0
2560 IF li=0 AND ac=0 AND na<>42 THEN PR
INT "Okay, ";ob$(na);" dropped.":ob(na)=
cp:zz=zz-1:GOTO 10
2562 IF na=42 AND cp<18 THEN np=cp+2:GOT
O 2570
2564 IF na=42 THEN np=cp-2:GOTO 2570
2566 na=li-12:IF na=24 THEN na=26
2568 li=0:ac=0:GOTO 2556
2570 PRINT "Boinnnggg!!!":ob(42)=np:zz=z
z-1:GOTO 10
```

Figure two

```
2550 IF na$="all" THEN 2580
2580 FOR i=1 TO nn
2582 IF i=61 AND ob(61)=-1 THEN PRINT "Y
ou leave the dog.":ob(61)=cp:zz=zz-1:IF
ob(11)<>-2 THEN lo=0:GOTO 2588
2584 IF i=26 AND ob(26)=-1 THEN rw=0:GOT
O 2587
2585 IF i=42 AND ob(42)=-1 THEN PRINT "B
oinnnggg!!!":IF cp<18 THEN np=cp+2:GOTO
2590
2586 IF i=42 AND ob(42)=-1 AND cp>17 THE
N np=cp-2:GOTO 2590
2587 IF ob(i)=-1 THEN PRINT "Okay, ";ob$
(na);" dropped.":ob(na)=cp:zz=zz-1
2588 NEXT
2589 GOTO 10
2590 ob(42)=np:zz=zz-1:GOTO 2588
```

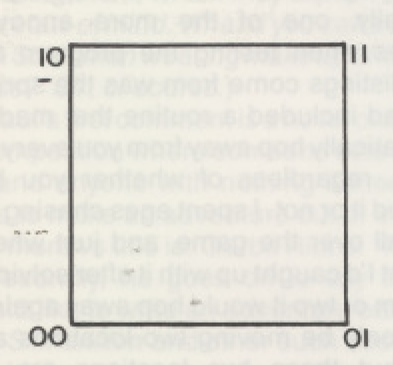
From Gray to black and white

Gordon Lee and Graham Barber try to harness Gray codes

A bonus for compiling the competitions takes the form of letters (mostly complimentary) which readers include with their entries. One such comes from Graham Barber of Sutton Coldfield, who writes:

"Enclosed is an idea for the Competition Page. During the past 20 months or so I've had so much pleasure and interest from DU (especially the competition) that I thought it time to put something back, and then being called a 'stalwart' in the June issue stirred me into action so you could say this is a 'Stalwart's revenge'."

Figure one



Graham's competition problem is to devise a program listing which will convert an input 'Gray' code into its binary equivalent. But, I hear you ask, what is a 'Gray' code? (*You can hear me in Birmingham!!?? — Ed.*)

This, at its most elementary, is a system of counting in which consecutive numbers are obtained by changing one digit and one digit only at each stage, the difference in that digit being 1. They were first developed by Frank Gray, an American research physicist, and the codes now bear his name. Gray codes can be formulated for number systems in any base — for example, in the familiar decimal system an adjacent run of codes might be 475, 476,576,566,466, .. etc. Note that there is no logical key to this sequence except that only one digit differs at each step. Because of this it is necessary to introduce certain additional rules if a logical sequence of numbers is to be formed.

But what is the point of all this? Consider the practical application of reading, say, a gas meter. With a conventional mechanical readout device there is no real problem when the reading changes from 10099 to 10100 even though three of the 'wheels' change simultaneously. But consider the situation with high speed machinery where rapid sampling needs accurate split second monitoring. The fewer digits changing between successive counts will reduce the possbility of errors. A typical link between a mechanical device and an electronic counter is by means of an optical disc using on/off switching and hence a binary system, so we will confine our attention to the *binary* Gray code. This was first developed by Frank Gray in the 1940s to reduce errors in the transmission of signals by pulse code modulation.

At first sight it might appear difficult to rationalise a logical order for binary Gray codes. If we limit ourselves to just two bits we get the cyclic series 00, 01, 11, and 10. This can be represented diagramatically by placing these four codes at the corners of a square (**figure one**). In this case each step in the sequence is forced, producing a single path (or its reverse) going around the square. Increasing the number of bits to three is equivalent to using the corners of a cube. Here the number of possible paths increases, as at each corner there is a choice of two possible routes. The only requirement is that the route chosen should take us to every corner once only. Starting at corner '000', it is quite easy to fine a number of possible routes, all different from each other, but nevertheless all producing a series of codes which obey the basic requirement. Some of the routes are cyclic, ie ending on an adjacent corner to '000'; others are not. For most applications, a cyclic route is preferable. For each additional digit used in a sequence, the number of possible paths increases at an alarming rate. Using just four digits, there are 5712 different sets of codes of which 2688 are cyclic. However, we are getting a bit off the track!

To be of practical use it is therefore necessary to formulate some additional rules whereby each binary Gray code will bear a 1:1 relationship with a conventional binary number. In this way conversion from one code to the other (in either direction) is

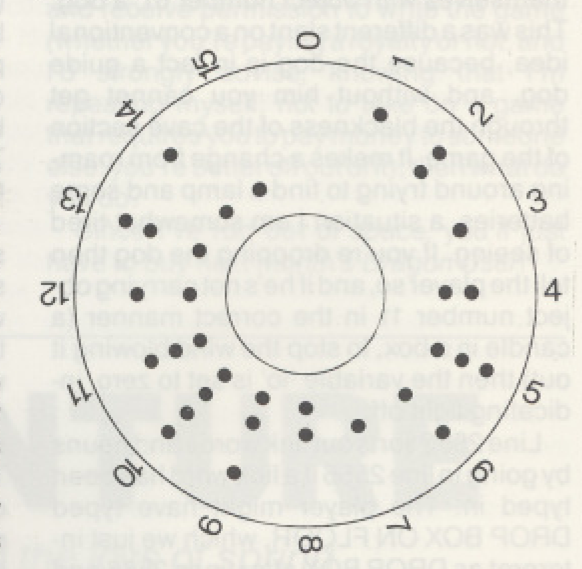


Figure two: using a 2-bit code to mark disc sectors.

Prize

IF the Dragon is the Computer that Time Forgot, then *Zotoka* is the land the Dragon remembered. Reviewed this month by Donald Morrison, *Zotoka* is the island paradise discovered by R & AJ Preston (currently nipping off to Malta to recharge their island-spotting batteries), which is invaded by hurricanes and mercenaries while you are in the presidential chair (Zotoka, not Malta). 10 paradise islands for 10 lucky winners. Hurricanes optional.

Rules

WHEN the wheel ceases to spin and your mind ceases to whirl, you may discover that your Dragon has produced an answer. Print it out, quickly add any notes or words of advice you deem necessary, send it to us in an envelope marked MARCH COMPETITION at the usual address, sit back and wait.

Oh yes, and don't forget the tiebreaker. We want to know why *you* think that *you* deserve to be sent to a paradise island. We'll be trying the lines out on the boss, who's heard it all, so make it good!

December winners

IT seems that falling into a black hole is good for the brain, because we had stashes of entries. The winners, who will get either a *Total Eclipse* or a discount voucher from John Penn Discount Software, are: Ian Huggins of Caerphilly, Clive Scott of Ashstead, R H Wilson of Basingstoke, Dave Lardener of Rutherglen, D R Sharples of Merseyside, Les Simpson of Littleport, Patricia Hill of Carshalton Beeches, John Blatch of Addlestone, Paul Priestland of Lechlade, James Bonfield of Sandy, J. Smallwood of Preston, Olaf Friis Neilsen of Denmark, Denis O'Mulloy of Cambridge, M. Busfield of Llandeyrn, Neil Davidson of Akrotiri, Christopher James of Malta, Don Robertson of Epsom, E A Newman of Addlestone, J. Smith of Twyford and S R Greenard of Ingatestone.

Favourite tiebreaker from Paul Priestland was "I fell into a black hole, and when I came out ... my Dragon User had actually arrived." This is an unreal universe we live in.

Solution

This month's solution is opposite.

a simple matter. The table shows the first few Gray codes together with their binary and decimal equivalents. **Figure two** shows a diagrammatic representation of an optical disc based on these numbers, each sector of the disc differing by just one bit from each of its neighbours. This is equivalent to the four-bit series given in the table. Reading from the centre of the disc outwards a hole represents a 1 and a non-hole, a zero. A similar disc constructed using conventional binary code would not work in the same way as there would be a difference in two of the bits when going from (decimal) 1 and 2, three bits between 3 and 4, and all four bits between 7 and 8.

DEC	BINARY	GRAY	DEC	BINARY	GRAY
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000
			0	0000	0000

This then is the basis of Graham Barber's competition question. Can you discover an easy method of converting from a binary Gray sequence to its equivalent standard (not BCD) binary number? Having done this, then adapt the algorithm into a computer program so that an input Gray code is converted and displayed. The method should be capable of converting any code of any reasonable length.

Hint: a binary and its Gray code equivalent will contain the same number of bits.

As a test of your technique, Graham also asks you to give the binary equivalents of Gray codes 11111111 and 101010101.

The Answer

This is Gordon Lee's own solution

see page 26 for results

ANSWER: The next cuboid in the series is one with sides 162, 163 and 164 units in length. This produces 4,330,584 individual unit cubes of which 4,173,120 form the central unpainted core. These painted blocks can be arranged into a perfect cube 54 units along each edge.

SOLUTION: Any cuboid h*w*l in dimensions will have a central inner core of h-2*(w-2)*(1-2). The difference between these two volumes will give the number of cubes in the single layer outer shell of the cuboid. These will be the cubes which bear paint on at least one of their faces.

In the listing H1 is the smallest edge of the cube (which must have a minimum size of 3 if there is to be any inner cube). From this dimension W1 and L1 are found. The corresponding dimensions of the inner cube are then calculated in line 40. The two respective volumes V1 and V2 are then found, and their difference (line 60) results in the number of unit cubes in the outer shell. All we need to do now is to print out those series of values in which this difference (P) is a perfect cube.

If you are a regular reader of these competitions you will at once realise that this is not a straightforward task as the cube root calculations on most micros is a bit of a hit and miss affair! Unfortunately, due to the logarithmic method of calculation used by the micro, cube roots (and other powers) can be a tiny bit out. This normally does not show on displayed results as these are correct to the nine digit value which is printed on screen. However, the computer's internal registers hold the value to several more significant figures and there may be a slight inaccuracy.

```
  10 H1=3
  20 W1=H1+1:L1=W1+1
  30 V1=H1*W1*L1
  40 H2=H1-2:W2=W1-2:L2=L1-2
  50 V2=H2*W2*L2
  60 P=V1-V2
  70 C=P^(1/3)
  80 C=INT(C):D=C+1
  90 IF C*C*C=P OR D*D*D=P THEN 1000
 100 H1=H1+1:GOTO 20
1000 PRINT H1;"   ";P
1010 GOTO 100
```

For example, if a value x is cubed and the cube root is calculated by the computer, the value that we would expect in a perfect world would be x. Unfortunately, the number that we do get is usually a minute fraction too high or too low. The program overcomes this by first finding the integral value of the number being tested. This will equal x if the root value was too high or x-1 if it was too low. Lines 80 and 90 of the program take these two values and find the cube of them by direct multiplication. If either of these values equals the number being tested then the result is printed out, as a perfect cube has been found. In the case of non-perfect cubes, neither value would be equal to the number under test.

This technique forms a much quicker alternative to merely testing by a succession of perfect cubes.

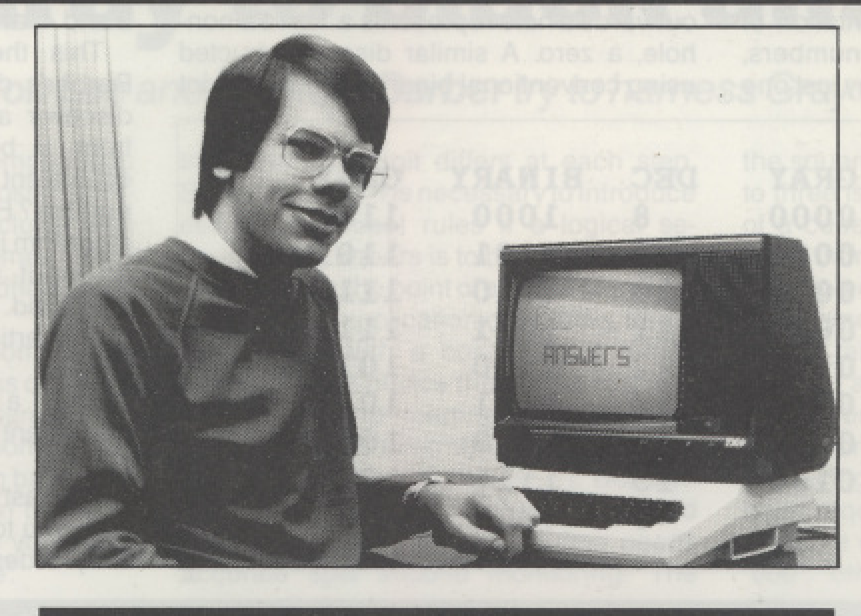
Dragon Answers

If you've got a technical question write to Brian Cadge. Please do not send a SAE as Brian cannot guarantee to answer individual inquiries.

Black to green again

WAY back in July 1983, you published a program in *Popular Computing Weekly* to invert text screen to green on black. I've used this in my program since then without any problems until now. The program I am writing uses the GET and PUT commands. No matter what I do the GET command causes a SN ERROR when used with the invert program.

K.Redhead
21 Baxter Avenue
Newcastle-Upon-Tyne
NE4 9QD



THE 'hook' used to interrupt the CLS command at location 416-418 is shared by the GET command. However, my program did not take account of this and this is what causes the SN ERROR. The simple solution is to surround the GET command with the POKES....

POKE 416,57 :GET(X,Y)-(W,Z),A,G: POKE 416,126

RAM and ROM

I had quite a response from the letter headed 'ROM can't be RAM' in the January '88 edition. As there seems to be quite an interest in a cartridge based RAM extension for the Dragon 32 here is some information on how to build your own.

The neatest solution was sent in by Mr.A.N Martin of 143 Black Haynes Road, Selly Oak, Birmingham B29 4RE. It is on his circuit that I have based the following article. Thanks to other readers who also sent in suggestions.

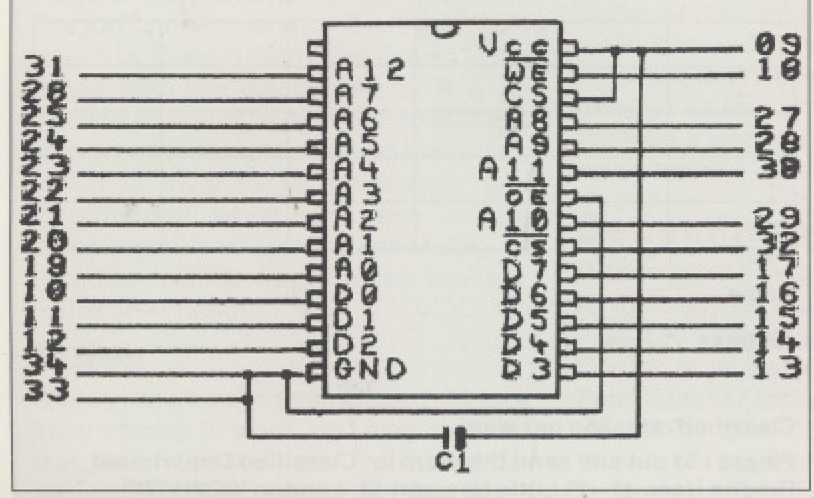
For an 8K RAM pack you'll need the following parts, all available from Maplin Electronics. (*Try 0702 554155 for enquiries*): a 6264 CMOS RAM chip, a 4.7mF capacitor (electrolytic), and a 28 pin IC socket for the RAM chip. You'll also need a prototyping board with 40-way edge connector and plastic case (available from Compusense). This costs about £10.

The diagram shows the RAM chip and its pin connections. The numbers at the end of the lines refer to the Dragon edge connector pin. The smoothing capacitor is connected between +5V and Ground and the 'CS' line is tied high. All other connections are direct to the edge connector.

The Dragon's edge connector is numbered as follows — looking at it end on the odd numbered pins 1,3,5 to 39 are above the board running right to left. The even numbered pins 2,4,6 to 40 are below the boards running right to left.

This is perhaps not a project for those who don't know one end of a soldering iron from another. But providing you take reasonable care and test your circuits with an ohm-meter (before connecting to the power!) then it is relatively straightforward. Take special care with the RAM chip as CMOS chips aresensitive to static! Touch something earthed, like a metal chair, before handling it.

The circuit does not decode the cartridge address space using A12-A15, but rather uses the CTS (Cartridge address space select) pin on the edge connector. This shouldwork on the Dragon 32, but the Dragon 64 might reqquire more rigorous address decoding.



6264 cmos ram chip, Dragon 32 8K cartidge ram.

PIN #	SIGNAL	DESCRIPTION
1	-12v	-12 Volts
2	+12v	+12 Volts
3	HALT	Halt input to the CPU (Inverted)
4	NMI	Non-Maskable Int. CPU (Inverted)
5	RESET	Main Reset Signal (Inverted)
6	E	Main CPU Clock (Ø.89 MHz)
7	Q	Quadrative Clock Signal
8	CART	FIRQ to CPU (Inverted)
9	+5v	+5 Volts
1Ø	DØ	CPU Data Bit Ø
11	D1	CPU Data Bit 1
12	D2	CPU Data Bit 2
13	D3	CPU Data Bit 3
14	D4	CPU Data Bit 4
15	D5	CPU Data Bit 5
16	D6	CPU Data Bit 6
17	D7	CPU Data Bit 7
18	R/W	CPU Read/Write Signal
19	AØ	CPU Address Bit Ø
2Ø	A1	CPU Address Bit 1
21	A2	CPU Address Bit 2
22	A3	CPU Address Bit 3
23	A4	CPU Address Bit 4
24	A5	CPU Address Bit 5
25	A6	CPU Address Bit 6
26	A7	CPU Address Bit 7
27	A8	CPU Address Bit 8
28	A9	CPU Address Bit 9
29	A1Ø	CPU Address Bit 1Ø
3Ø	A11	CPU Address Bit 11
31	A12	CPU Address Bit 12
32	CTS	Cartidge Select Signal (Inverted)
33	GND	Signal Ground
34	GND	Signal Ground
35	SND	Sound Input to Multiplexor
36	SCS	Alternative Select (Inverted)
37	A13	CPU Address Bit 13
38	A14	CPU Address Bit 14
39	A15	CPU Address Bit 15
4Ø	SLENB	Input to disable device selection

I/O as input

COULD you tell me the I/O address of the Dragon printer port and how to program it as Input, rather than Output?

Richard Evans
St.Austell
Cornwall

YOU are correct in assuming that the printer port is connected to a PIA. Its address is $FF02 for the data register and $FF03 for the control register. However, the 8 data lines are shared with the Dragon's keyboard and so it is not possible to use the port programmed as input. Instead you'll have to obtain one of the I/O ports available which plug into the cartridge socket.

Any comments on this circuit are welcome. Mr. Martin had yet to fully build and test a version of his RAM pack and has tentatively offered to build one for a *Dragon User* reader for a tenner — but please contact him first!